

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA



Trabajo Fin de Máster

**Implementación de redes neuronales para
conducción autónoma**
(Implementation of neural networks for
autonomous driving)

Para acceder al Título de

***Máster Universitario en
Ingeniería de Telecomunicación***

Autor: Alejandro Saiz Allende

Marzo - 2021

MASTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

CALIFICACIÓN DEL TRABAJO FIN DE MASTER

Realizado por: Alejandro Saiz Allende

Director del TFM: Adolfo Cobo García

Título: “Implementación de redes neuronales para conducción autónoma”

Title: “Implementation of neural networks for autonomous driving”

Presentado a examen el día:

para acceder al Título de

MASTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

Composición del Tribunal:

Presidente (Apellidos, Nombre): Mauro Lomer Barboza

Secretario (Apellidos, Nombre): José Angel Miguel Díaz

Vocal (Apellidos, Nombre): Rafael Menéndez del Llano Rozas

Este Tribunal ha resuelto otorgar la calificación de:

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del TFM
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Trabajo Fin de Máster N°
(a asignar por Secretaría)

Agradecimientos

A mi familia, amigos y profesores por apoyarme y ayudarme.

Resumen

La inteligencia artificial y el aprendizaje automático se han convertido en el foco de atención de desarrolladores e investigadores en los últimos años, destacando la rama enfocada a las técnicas de aprendizaje profundo o *deep learning*, que han revolucionado campos como la genética, visión artificial, sistemas de antivirus o sistemas de conducción autónoma entre otros, en todos ellos ha logrado resultados muy superiores a los obtenidos con técnicas tradicionales.

A lo largo de este documento se profundizará en el último de los campos comentados, el de la conducción autónoma, aprendiendo sus características, como funciona y de que se compone un vehículo autónomo. Haciendo uso de técnicas de *deep learning*, en concreto redes neuronales convolucionales y de regresión logística, se creará un sistema software inteligente que permita a un vehículo saber cuando debe parar y cuando continuar la marcha, sirviendo de esta forma como la base de un sistema más complejo que desembocaría en un vehículo autónomo.

Palabras clave

Conducción autónoma, aprendizaje profundo, redes neuronales artificiales, redes neuronales convolucionales, redes YOLO, redes neuronales de regresión.

Abstract

Artificial intelligence and machine learning have become the focus of attention of developers and researchers in recent years, highlighting the branch focused on deep learning techniques, which have revolutionized fields such as genetics, artificial vision, systems antivirus or autonomous driving systems, among others, in all of them it has achieved far superior results than those obtained with traditional techniques.

Throughout this document, the last of the commented fields, that of autonomous driving, will be studied in depth, learning its characteristics, how it works and what an autonomous vehicle is made of. Using deep learning techniques, specifically convolutional neural networks and logistic regression, an intelligent software system will be created that allows a vehicle to know when to stop and when to continue moving, thus serving as the basis of a more complex that would lead to an autonomous vehicle.

Keywords

Autonomous driving, deep learning, artificial neural networks, convolutional neural networks, YOLO networks, regression neural networks.

Índice

| | | |
|-------|--|----|
| 1 | Introducción | 1 |
| 1.1 | Introducción | 1 |
| 1.2 | Objetivos y estructura | 2 |
| 2 | Estado del Arte | 3 |
| 2.1 | ¿Qué es la conducción autónoma? | 3 |
| 2.1.1 | Niveles de conducción autónoma | 3 |
| 2.2 | Tecnología implementada en el vehículo | 5 |
| 2.2.1 | Sensores principales | 6 |
| 2.2.2 | Otros sensores y sistemas | 8 |
| 2.2.3 | Redundancia de los sistemas | 11 |
| 2.2.4 | IoT y los vehículos autónomos | 12 |
| 2.2.5 | Fusión de sensores e Inteligencia Artificial | 13 |
| 2.3 | Inteligencia Artificial y Machine Learning | 15 |
| 2.3.1 | Redes Neuronales Artificiales..... | 15 |
| 2.3.2 | Paradigmas de Aprendizaje | 21 |
| 2.3.3 | Redes Neuronales Convolucionales | 23 |
| 2.3.4 | Redes Neuronales de Regresión..... | 26 |
| 3 | Implementación | 28 |
| 3.1 | Creación del dataset..... | 28 |
| 3.1.1 | Etiquetado con labellmg | 29 |
| 3.1.2 | División del dataset en grupos de entrenamiento y test | 30 |
| 3.2 | Red neuronal convolucional: YOLOv5 | 31 |
| 3.2.1 | Funcionamiento de una red YOLO | 32 |
| 3.2.2 | Transferencia de aprendizaje e implementación | 34 |
| 3.2.3 | Resultados y rendimiento | 34 |
| 3.3 | Red neuronal de regresión logística..... | 36 |
| 3.3.1 | Creación del dataset..... | 36 |
| 3.3.2 | Implementación y rendimiento..... | 38 |
| 4 | Sistema completo y resultados conjuntos | 41 |
| 4.1 | Resultados finales..... | 41 |
| 4.1.1 | Análisis de los fotogramas por segundo | 44 |
| 5 | Conclusión y líneas futuras..... | 45 |
| 6 | Referencias | 48 |

| | |
|--|----|
| Figura 1 Niveles de conducción autónoma | 4 |
| Figura 2 Sensores principales de un vehículo autónomo..... | 6 |
| Figura 3 Sistema de cámaras instalado en el vehículo..... | 7 |
| Figura 4 Mapeado del entorno del sistema LIDAR..... | 8 |
| Figura 5 Captura de un sensor infrarrojo | 9 |
| Figura 6 Comunicación V2V..... | 11 |
| Figura 7 Sistemas redundantes de un vehículo autónomo..... | 12 |
| Figura 8 Mapa cartografiado para conducción autónoma..... | 14 |
| Figura 9 ANN: Neurona | 15 |
| Figura 10 ANN: Estructura..... | 16 |
| Figura 11 ANN: Neurona con función de activación | 17 |
| Figura 12 Función de activación escalón..... | 17 |
| Figura 13 Función de activación sigmoide | 18 |
| Figura 14 Función de activación tanh | 18 |
| Figura 15 Función de activación ReLU..... | 19 |
| Figura 16 Aplicación del descenso del gradiente | 20 |
| Figura 17 Backpropagation en una red neuronal..... | 20 |
| Figura 18 Aprendizaje supervisado | 21 |
| Figura 19 Aprendizaje no supervisado | 22 |
| Figura 20 Aprendizaje reforzado | 23 |
| Figura 21 Aplicación del filtro en una CNN | 24 |
| Figura 22 Aplicación de un filtro sobre una imagen | 25 |
| Figura 23 Estructura de una red neuronal convolucional | 25 |
| Figura 24 Arquitectura de una red neuronal convolucional | 26 |
| Figura 25 Gráfica de una Regresión Logística | 27 |
| Figura 26 Imagen etiquetada con labelImg..... | 29 |
| Figura 27 Dataset completo etiquetado | 30 |
| Figura 28 Evaluación del dataset..... | 31 |
| Figura 29 YOLO: Imagen dividida en SxS..... | 32 |
| Figura 30 YOLO: Combinación de cuadros delimitadores..... | 32 |
| Figura 31 YOLO: Mapa de probabilidades..... | 33 |
| Figura 32 YOLO: Resultado final | 33 |
| Figura 33 Resultados YOLO: Precisión vs Épocas | 35 |
| Figura 34 Resultados YOLO: Imágenes etiquetadas..... | 35 |
| Figura 35 Dataset de la red de regresión en formato CSV | 37 |
| Figura 36 Exactitud de la red de regresión..... | 38 |
| Figura 37 Precisión de la red de regresión | 38 |
| Figura 38 Matriz de confusión de la red de regresión | 39 |
| Figura 39 Curva ROC de la red de regresión | 40 |
| Figura 40 Resultados finales - Secuencia de detección..... | 41 |
| Figura 41 Resultados finales - Frena | 42 |
| Figura 42 Resultados finales - Adelante..... | 42 |
| Figura 43 Resultados finales - Vía libre de objetos | 43 |
| Figura 44 Resultados finales - Mala detección..... | 43 |
| Figura 45 Jetson Nano de Nvidia..... | 47 |

| | |
|--|--------|
| Ecuación 1 Función Escalón..... | 17 |
| Ecuación 2 Función Sigmoide | 18 |
| Ecuación 3 Función tanh | 18 |
| Ecuación 4 Función ReLU | 19 |
| Ecuación 5 Formula de la precisión para una red de regresión | 38 |
| Tabla 1 Rendimiento del sistema con un video | 44 |

1 Introducción

1.1 Introducción

El concepto sobre vehículos que sean completamente autónomos y no requieran de la interacción humana ha hecho volar la imaginación del ser humano durante años. En series o películas de ciencia ficción como Batman, Minority Report o Total Recall entre otras muchas, se ha fantaseado con este tema, pero en los últimos años se vislumbra un futuro donde esta fantasía de ciencia ficción es más real que nunca.

A día de hoy existen vehículos con capacidad de conducción autónoma, como pueden ser los Tesla con su sistema de AutoPilot, pero siguen requiriendo de la presencia de un conductor humano dado que, en muchos países este tipo de tecnologías de conducción completa o parcialmente autónomas no están permitidas.

Pasar de un vehículo normal a un vehículo autónomo implica añadir a este una gran cantidad de sensores, tales como cámaras, sistema radar, detectores de posición, entre otros. Toda la información generada por estos sensores es la que emplea un sistema informático para decidir cómo y hacia donde mover el vehículo. Esto es posible gracias a la inteligencia artificial, que hace al sistema informático capaz de entender y comprender toda la información recopilada para que pueda gobernar de forma eficiente y segura el vehículo.

La inteligencia artificial abarca una extensa parte dentro de las Ciencias Computacionales, por no decir que casi todas las ramas la implementan de alguna forma. Pero, aunque pueda parecer un concepto muy novedoso, realmente la idea de una máquina con capacidad para pensar fue propuesta por Alan Turing en 1950, quien a su vez ideó un método para juzgar si las respuestas de una máquina son indistinguibles de las de un ser humano, conocida como la prueba de Turing o el Test de Turing.

Es en 1956 en la conferencia de Dartmouth en Estados Unidos cuando se acuña el término inteligencia artificial por John McCarthy, Marvin Minsky y Claude Shannon, donde la definieron como *“la ciencia e ingenio de hacer máquinas inteligentes, especialmente programas de cálculo inteligente”*. A pesar que en ese momento se hicieron previsiones triunfalistas sobre este campo, estas no llegaron a cumplirse provocando el abandono casi total sobre este campo de investigación.

Ha sido hace unos años y gracias a la evolución y mejora de la tecnología y el desarrollo de nuevas técnicas que la inteligencia artificial se ha empezado a utilizar de forma masiva en una gran cantidad de campos como por ejemplo el reconocimiento de imágenes y caracteres, de voz, predicciones bursátiles, conducción autónoma, generación de texto, análisis genético, traducción de idiomas, etc.

Este proyecto aplicará en concreto técnicas de reconocimiento de imágenes y análisis de datos aplicadas al mundo de la conducción autónoma empleando como punto de salida imágenes y videos reales.

1.2 Objetivos y estructura

El objetivo de este proyecto es comprender como funcionan los vehículos autónomos, centrándose en la inteligencia artificial que incorporan. Para ello ha sido necesario desde aprender nuevos lenguajes de programación hasta las técnicas y modelos para la creación y uso de redes neuronales completamente funcionales, en este caso el proyecto se basa únicamente en el uso de redes neuronales convolucionales y redes de regresión.

El programa software analizará las imágenes que captaría una cámara situada en la parte frontal del vehículo. El programa será capaz de diferenciar vehículos, personas y pasos de cebrá y además enviar un aviso indicando si el vehículo debe frenar o puede continuar el recorrido, en función de la cercanía de alguno de los elementos detectados.

El proyecto se divide en cuatro apartados, siendo el primero una introducción a la situación del vehículo autónomo en la actualidad. Se explicará que se entiende por conducción autónoma y cuantos grados o niveles tiene, también se expondrán todos los sensores que puede incorporar a día de hoy un vehículo de estas características y que función específica cumple cada uno de ellos dentro del vehículo. Vinculado a esto también se hablará de la integración de vehículos autónomos dentro del mundo de las telecomunicaciones, en concreto dentro del campo del Internet de las Cosas, también conocido como IoT de las siglas en inglés.

Dentro de este mismo apartado se realizará una exposición sobre la inteligencia artificial y el *machine learning*, explicando en detalle que es una red neuronal artificial y como funciona. El apartado finalizará hablando sobre las dos redes empleadas en el proyecto, la primera una red convolucional y la segunda una red de regresión logística.

El segundo apartado detalla todo el desarrollo seguido paso a paso para la implementación de las dos redes neuronales que conforman el sistema de detección, así como el funcionamiento de ambas. Adicionalmente se presentará el rendimiento individual de cada una de las redes.

En el tercer apartado se divide a su vez en dos partes diferenciadas, en primer lugar, se explicará cómo se han unido las dos redes para que trabajen de forma conjunta buscando un uso óptimo de los datos de entrada. En segundo lugar, se recopilan todos los resultados obtenidos en las simulaciones finales del sistema completo, presentando todos los casos que se han dado y analizando su rendimiento al trabajar con videos, ya que el sistema está pensado para trabajar en tiempo real con las imágenes captadas con la cámara.

Finalmente, el cuarto y último apartado expone una conclusión final sobre el proyecto, así como posibles líneas futuras tanto de investigación como de mejora para el mismo.

2 Estado del Arte

2.1 ¿Qué es la conducción autónoma?

La conducción autónoma, como propiamente indica, persigue la idea de un coche que conduzca por nosotros, al cual solo haga falta decirle a dónde queremos ir y él nos lleve hasta allí de forma segura, mientras nosotros podemos despreocuparnos sobre todos los aspectos relacionados con la conducción. Si bien es cierto que a día de hoy no existen coches 100% autónomos, cada vez son una realidad más posible gracias al avance de las telecomunicaciones y el desarrollo de la Inteligencia Artificial, ambas trabajando en conjunto con una amplia gama de sensores.

Una de las principales motivaciones para el desarrollo de coches autónomos es la seguridad en la carretera, puesto que cada año se registran miles de muertes 2 en accidentes de tráfico y un número de heridos aún mayor, según un estudio de la Unión Europea el 95% de estos incidentes se deben a un error humano.

Otro aspecto que aún está por resolver en cuanto a los coches autónomos es su legislación, ya que hay muchas dudas, por ejemplo, sobre cómo debe aplicarse la ley en caso de un accidente, ¿Quién sería considerado culpable? También surge aquí otra cuestión, ¿podrían convivir estos coches autónomos con otro tipo de vehículos?

A día de hoy los coches incluyen una gran cantidad de sensores que aumentan la seguridad al volante, sistemas de pre-colisión, sensores de cambio involuntario de carril, control inteligente de las luces, reconocimiento de señales de tráfico, etc. Estos sistemas hacen que el vehículo sea autónomo en ciertos aspectos, aunque el conductor siempre tiene el control sobre las decisiones finales. Un coche autónomo también podría permitir a personas con dificultades para conducir, poder emplear un coche propio para desplazarse, sin depender de nadie más.

En base a esta información se puede intuir que hay diferentes niveles o grados de automatización posibles en un vehículo, en este caso se estudiarán los relativos al proceso de conducción autónoma en sí mismo, estos niveles son definidos a continuación.

2.1.1 Niveles de conducción autónoma

Estos niveles se determinan en función del grado de automatización en la conducción que ofrece un vehículo, diferenciando a tres actores principales en la conducción: el conductor (humano), el sistema de automatización y otros sistemas y componentes del vehículo. Los niveles se definirán en función de la labor que desempeñe cada uno de estos actores durante la tarea de la conducción dinámica o DDT de las siglas en inglés (*Dynamic DrivingTask*).

Los sistemas de seguridad activos, como el control electrónico de la estabilidad o el frenado de emergencia automático y ciertos tipos de sistemas de asistencia al conductor, como el aviso si se va a salir del carril, se excluye del grupo de elementos a considerar para determinar el grado de automatización en la conducción de un vehículo, ya que no realizan parte o todo el proceso de DDT de forma sostenida, sino más bien realizan intervenciones momentáneas en situaciones concretas. Estos sistemas pueden incluirse en vehículos de cualquier nivel de automatización en la conducción

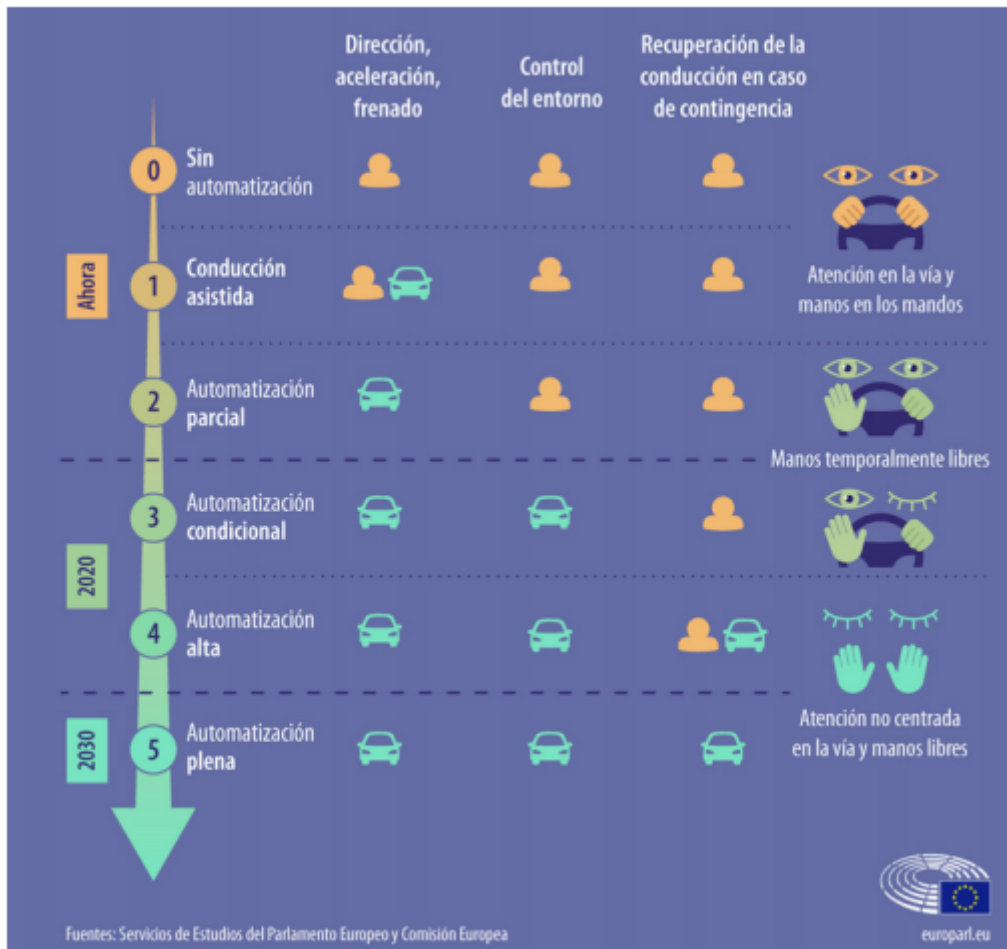


Figura 1 Niveles de conducción autónoma

2.1.1.1 Nivel 0: Sin automatización en la conducción

En este nivel todas las acciones las realiza en su totalidad el conductor humano, que es el único responsable, es el método de conducción tradicional.

2.1.1.2 Nivel 1: Asistencia en la conducción

El vehículo cuenta con algún sistema de ayuda durante la conducción, como podrían ser los sistemas de control de velocidades adaptativos, que buscan hacer la conducción más cómoda a la vez que se incrementa la seguridad. El conductor humano sigue siendo el único responsable del control y la toma de decisiones.

2.1.1.3 Nivel 2: Automatización parcial

En este caso el vehículo cuenta con un sistema de control de movimiento tanto longitudinal como lateral, aunque no tiene mecanismos de detección y respuesta ante objetos. Es necesario que un conductor humano esté presente en todo momento.

El vehículo tendrá la capacidad para actuar de forma independiente, pudiendo replicar varias tareas ordenadas por el conductor, como podría ser un sistema de aparcado automático, ayudas

en el mantenimiento de la trayectoria o asistencia a la frenada. En este nivel la seguridad al volante se ve incrementada en gran medida.

2.1.1.4 Nivel 3: Automatización condicionada

La autonomía de los vehículos es más elevada, aunque se sigue requiriendo de un conductor que esté atento para intervenir en caso de ser necesario. En este nivel aparte de contar con los sistemas de movimiento longitudinal y lateral, se cuenta con sistemas de detección y respuesta ante objetos. El vehículo será capaz de analizar su entorno para poder actuar o tomar decisiones que al conductor no le da tiempo a realizar o no ha llevado a cabo por un despiste.

El vehículo podrá decidir, por ejemplo, cuando cambiar de carril o cuando frenar para evitar una colisión con otro vehículo.

2.1.1.5 Nivel 4: Automatización elevada

En este nivel no se requiere de una intervención humana en ningún momento de la conducción, el vehículo será quien lo controle todo, el tráfico, los objetos en la vía, las condiciones del entorno, definirá él la ruta a seguir y responderá ante las situaciones que se encuentre, realizando una lectura constante del entorno que lo rodea.

El controlador humano solo tendrá que decir el lugar de destino y el vehículo lo llevara hasta allí, sin embargo, si el sistema fallase, se requerirá que el conductor humano tome el mando, el vehículo cuenta con todos los sistemas necesarios para el control manual. El conductor humano puede tomar también el control del vehículo cuando el desee, el sistema no bloquea en ningún momento estos controles.

2.1.1.6 Nivel 5: Automatización completa

Este es el nivel máximo de automatización posible para un vehículo, este no necesitará contar con los elementos de control tradicionales (volante, pedales, etc.), el usuario se comunicará con él a través de una interfaz donde solo tendrá que seleccionar el destino. En caso de fallo, el vehículo dispondrá de un sistema de respaldo que tome el control en caso de cualquier imprevisto.

Aquí la figura del conductor ya no existe, así como la función de conducir como tal, el usuario podrá dedicarse a hacer otras cosas de camino a su destino. Las órdenes para el guiado podrán darse a través de la voz, empleando una interfaz o un dispositivo móvil.

2.2 Tecnología implementada en el vehículo

Los sistemas de conducción autónoma requieren una serie de subsistemas y conjuntos de sensores interconectados entre sí para poder obtener información precisa del entorno que rodea al vehículo, también le permiten procesar toda esta información para crear la mejor ruta posible del punto A al punto B.

Cada tipo de sensor tiene sus propias fortalezas y debilidades en términos de alcance, capacidad de detección o confiabilidad. Los sistemas de conducción autónomos emplean una gran cantidad de tecnologías para proporcionar redundancia al sistema completo y así intentar prever posibles fallos.

Además, toda esta información recopilada puede compartirse con otros vehículos que estén conectados entre sí, de forma que se pueda encontrar la mejor ruta, prevenir atascos o esquivar cualquier obstáculo en la carretera.

Se denomina fusión de sensores al proceso por el cual se pone en común toda la información recogida por los diferentes sensores para poder procesarla.

2.2.1 Sensores principales

Los vehículos autónomos emplean una gran cantidad de sensores de diversos tipos, cada uno empleado para una serie de funciones concretas, a continuación, se exponen los tres sensores que actualmente todos los fabricantes de vehículos autónomos implementan: cámaras, radares y la unidad LIDAR.

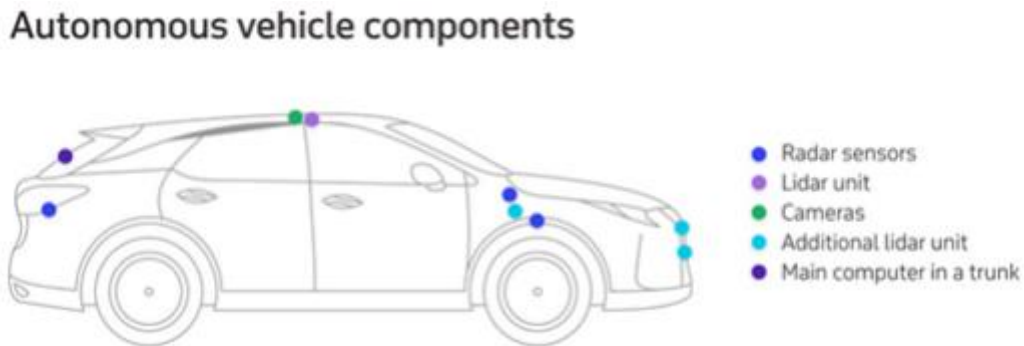


Figura 2 Sensores principales de un vehículo autónomo

2.2.1.1 Sistema de cámaras

Los vehículos incorporan una serie de cámaras que realizan la función del ojo del conductor humano durante la conducción, pero con un rango de 360°. De esta forma pueden obtener información detallada de su alrededor, la cual procesará el software del sistema, que posteriormente sea el que decida la realización de ciertas acciones en base a lo que ve en las imágenes. La efectividad de las cámaras se reduce cuando hay poca luz o cuando la climatología es adversa.

Hoy en día pueden emplearse cámaras 3D que consiguen imágenes más detalladas y realistas, estos sistemas pueden detectar objetos y clasificarlos, ver si se trata de otro vehículo, de un peatón, una señal, etc. Son capaces de determinar las distancias entre los objetos que ven.

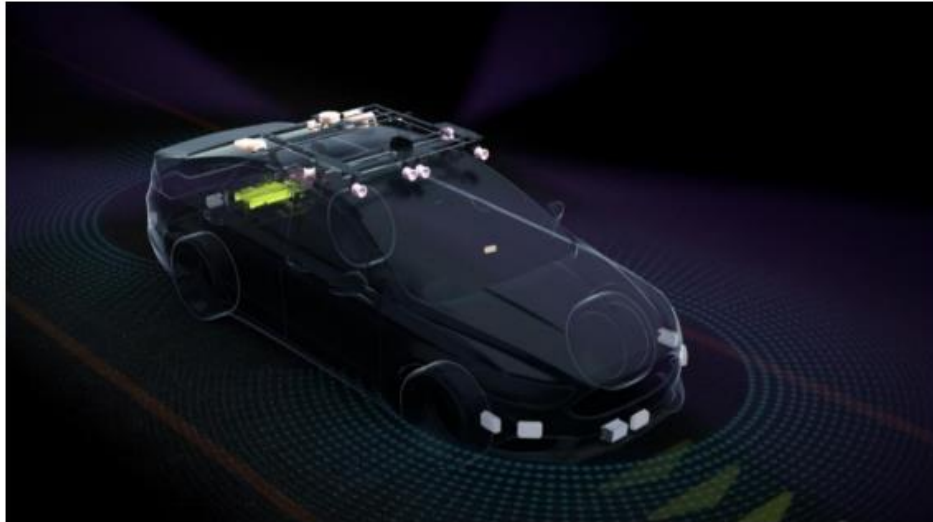


Figura 3 Sistema de cámaras instalado en el vehículo

Lamentablemente este tipo de sensores tienen dificultades para localizar objetos cuando las condiciones climáticas son adversas, el sistema de procesamiento no puede trabajar con estas imágenes de tan baja calidad.

2.2.1.2 Radar

Este sistema emplea ondas electromagnéticas para detectar objetos y poder realizar un seguimiento de los mismos, ya que no solo permite detectar un objeto, sino también su dirección y velocidad en caso que se esté moviendo en tiempo real. Ofrece una gran precisión para distancias de hasta 300 metros incluso en condiciones climatológicas adversas, se suelen implementar dos tipos de radares, uno de corto alcance y otro de largo alcance.

- El radar de corto alcance (24 GHz) permite el monitoreo del punto ciego del coche, asistencia para el aparcado o sistemas para indicar que el vehículo está en su carril.
- El radar de largo alcance (77 GHz) se emplea para el control automático de distancia entre vehículos y la asistencia de frenado.

Por lo general los radares son de tipo 2D, solo realizan un escaneo horizontal, debido a esto no pueden determinar la altura de los objetos y presentan problemas a la hora de identificar peatones, señales de tráfico o al conducir debajo de puentes. Actualmente se están desarrollando una variedad de radares 3D que resolverán estos problemas.

2.2.1.3 LIDAR (Laser Imaging Detection And Ranging)

Este sistema funciona de forma similar a un radar, pero emplea láseres de tipo pulsado en lugar de ondas de radio, puede calcular la distancia que hay entre un emisor laser y un objeto. Los rayos emitidos por estos láseres son seguros para los ojos de las personas.

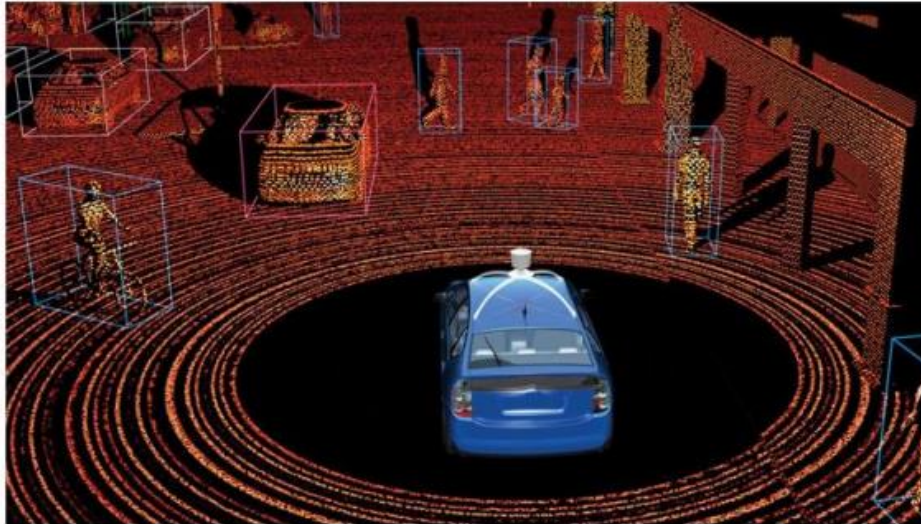


Figura 4 Mapeado del entorno del sistema LIDAR

La información obtenida por estos sistemas complementa a la obtenida por el sistema radar, aportando información extra muy detallada sobre el entorno que permite crear una imagen 3D de su alrededor. Se puede configurar para crear un mapa completo de 360º alrededor del vehículo en lugar de depender de un campo de visión estrecho.

Estos mapas tienen especial relevancia para los sistemas autónomos de los niveles 4 y 5.

Al igual que ocurría con las cámaras, estos láseres presentan malos resultados en situaciones de niebla o nieve, donde se puede bloquear el sistema emisor y este no ser capaz de detectar los objetos que hay en la carretera. Otro inconveniente con este sistema es su elevado precio.

2.2.2 Otros sensores y sistemas

Los tres sensores mencionados en el apartado anterior son los más importantes para la automatización de la conducción, pero un coche autónomo puede implementar aún más sensores con la finalidad de obtener información más precisa empleando sensores complementarios.

2.2.2.1 Sensores Ultrasónicos

Se emplean para la detección de objetos que se encuentren a poca distancia del vehículo, forman parte del sistema de ayuda para el aparcado. Funcionan emitiendo ondas sonoras no audibles y calculando el tiempo que tardan estas en volver al punto de emisión.

2.2.2.2 Sensores Infrarrojos

Estos sensores se utilizan para realizar un seguimiento de objetos cuando no hay mucha luz.

Actualmente se están diseñando cámaras de infrarrojo lejano (también conocidas como cámaras térmicas) que detectan longitudes de onda por debajo del espectro visible que indican calor. Estas permiten detectar a seres vivos en situaciones adversas, como niebla, lluvia abundante o nieve. También permiten la detección en la oscuridad.

La siguiente imagen tomada de noche muestra como gracias una cámara de este tipo permite distinguir los coches detenidos en la carretera, las personas que están cruzando la misma y las diferentes señalizaciones de tráfico y semáforos.



Figura 5 Captura de un sensor infrarrojo

2.2.2.3 Sistemas de navegación inercial

También conocidos como IMU (*Internal Measurement Unit*), emplean sensores de movimiento (acelerómetros) y sensores giroscópicos para saber cuál es la posición el vehículo, cual es la dirección de su trayectoria y a qué velocidad va, este sistema no sabe dónde se encuentra el vehículo, solo su movimiento. El sistema de navegación inercial se emplea junto con los sistemas de posicionamiento para conseguir más precisión a la hora de situar al vehículo en el mapa.

Este sistema no necesita conexión o conocimiento del mundo externo, esta independencia convierte a la IMU en una tecnología central para la seguridad y desempeña un papel fundamental en la fusión de sensores.

La IMU permite dirigir al vehículo de forma dinámica, evitando que este deambule por el carril dando el aspecto que esta conducido por un mal conductor. Puede incluso dirigir al vehículo durante periodos cortos cuando alguno de los otros sensores se desconecta.

Otra función que desempeña este sistema tiene que ver con la seguridad, empleando sus giroscopios pueden implementar detectores de vuelco, que en caso de activarse podrían informar al sistema de comunicaciones para informar de un accidente.

Un problema que presentan estos sistemas, incluso los más modernos, es un leve error de cálculo que se va a acumulando en el tiempo. Cuando la IMU mide la aceleración del cuerpo, esta

se integra para obtener la velocidad y luego se integra una segunda vez para estimar la posición. La integración del error asociado a los sensores genera una deriva con el tiempo. Hasta la IMU más precisa puede generar un error de varios metros al cabo de 60 segundos.

2.2.2.4 Posicionamiento y navegación por satélite

El uso de estos sistemas en los automóviles está extendido desde hace varios años y permiten localizar al vehículo en un punto del planeta empleando una red de satélites con un margen de error de unos pocos metros. Este sistema es inmune a situaciones de climatología adversa y, pero presenta problemas cuando las señales transmitidas no llegan al satélite, por ejemplo, si el vehículo está atravesando un túnel o un paso subterráneo. En esta situación la IMU puede proporcionar una solución a corto plazo, ésta estimará la posición del vehículo hasta que se pueda acceder nuevamente a los satélites.

Existen varios sistemas de posicionamiento y navegación vía satélite en funcionamiento hoy en día, entre ellos se podría destacar:

- GPS (*Global Positioning System*): Desarrollado por el departamento de defensa de los Estados Unidos, emplea una red de 24 satélites.
- GLONASS: Desarrollado por la URRS y administrado actualmente por la Federación Rusa. Emplea una red de 31 satélites.
- Galileo: Desarrollado por la Unión Europea en conjunto con la Agencia Espacial Europea, cuando esté en pleno funcionamiento contará con 30 satélites, actualmente tiene 26 satélites en órbita.

Muchos vehículos llevan este tipo de sistemas integrados y disponen de un motor de cómputo específico para realizar el procesamiento de la información, solo necesitan disponer de una antena para poder funcionar.

2.2.2.5 Sistemas de comunicación

Gracias a los últimos avances en los sistemas de telecomunicaciones se ha hecho posible una comunicación directa y en tiempo real entre diferentes sistemas que involucran el proceso de la conducción autónoma. Este tipo de comunicaciones emplean principalmente dos tecnologías, 5G y DSRC (comunicación de corto alcance), ésta segunda opera a 5.9 GHz, tiene un alcance de un kilómetro aproximadamente y una latencia inferior a los 2 ms, su estándar es el IEEE 802.11p.

Aparece aquí el concepto V2X, que significa “vehículo para todo”. La información recopilada por los sensores y otras fuentes puede viajar a través de enlaces de gran ancho de banda, baja latencia y alta confiabilidad hacia un receptor, éstas pueden ser:

- V2V: vehículo a vehículo
- V2I: vehículo a infraestructura
- V2P: vehículo a peatón
- V2R: vehículo a red

La siguiente imagen presenta otra aplicación de esta filosofía, en ella se puede ver un coche que se ha parado para dejar cruzar la calle a un peatón, este coche envía esta información a otro coche que se aproxima hacia esa dirección, advirtiéndole.

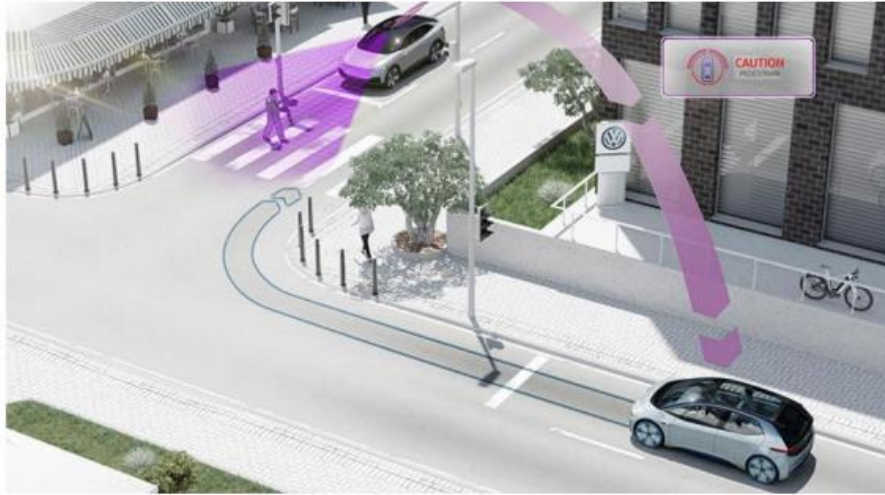


Figura 6 Comunicación V2V

El planteamiento de este ecosistema busca la conexión de todos los elementos de una vía entre sí, permitiendo a los vehículos comunicarse entre sí u obtener información sobre el entorno que los rodea, como puede ser comprobar el estado de un semáforo o mirar si hay aparcamiento libre en una zona y dónde está.

Permite también a los vehículos intercambiar información con las estaciones al borde de la carretera, está por lo general es relativa a la seguridad o las condiciones generales de la circulación, con el fin de evitar accidentes o congestiones en el tráfico. Un caso práctico podría ser el envío de alertas por límite de velocidad, pero también pagos electrónicos para peajes o estacionamiento.

La conducción autónoma necesita comunicarse con otros vehículos, pasajeros y el resto elementos involucrados durante la conducción. Así se puede determinar la posición exacta del vehículo durante su recorrido y éste puede decidir cómo comportarse en una situación dada, ya que cuenta con una gran cantidad de información extra sobre los tramos que están por venir.

2.2.3 Redundancia de los sistemas

Los sistemas de conducción autónoma deben tener un plan de respuesta en caso que ocurra cualquier error durante la conducción, ya que lo principal es mantener la seguridad de los pasajeros, así como la de las personas o vehículos del entorno y del propio vehículo. En los vehículos autónomos, más en concreto los de los últimos niveles, el conductor no siempre está atento a la carretera, por tanto, las funciones que desempeña el vehículo deben ser extremadamente confiables.

Para lograr esto es necesario añadir varios niveles de redundancia en todos los sistemas que incorpore el vehículo, en especial en los que desarrollen funciones de seguridad vitales, como sea el sistema de frenado o la dirección. Se generarán varias entradas de información en las que poder confiar, en lugar de solo una principal. De esta manera si algo falla y una entrada no funciona, hay otras a las que recurrir. Los sistemas de percepción, localización y planificación también requieren de redundancia.

Además, la redundancia genera seguridad en los consumidores, que se sienten más seguros gracias a la profunda complejidad que tienen estos sistemas que pueden llegar a manejar

cualquier situación. En los vehículos autónomos de nivel 4 o 5 la redundancia se vuelve aún más crítica, a medida que aumenta el intervalo de tiempo que el conductor no está prestando atención.

Este diagrama muestra de una forma conceptual como la gran mayoría de elementos del vehículo están duplicados, de forma que si uno falla se activa el de respaldo de forma inmediata y así se mantiene en perfecto funcionamiento los sistemas hasta llegar a un punto de reparación donde arreglar el componente o parte que ha fallado.

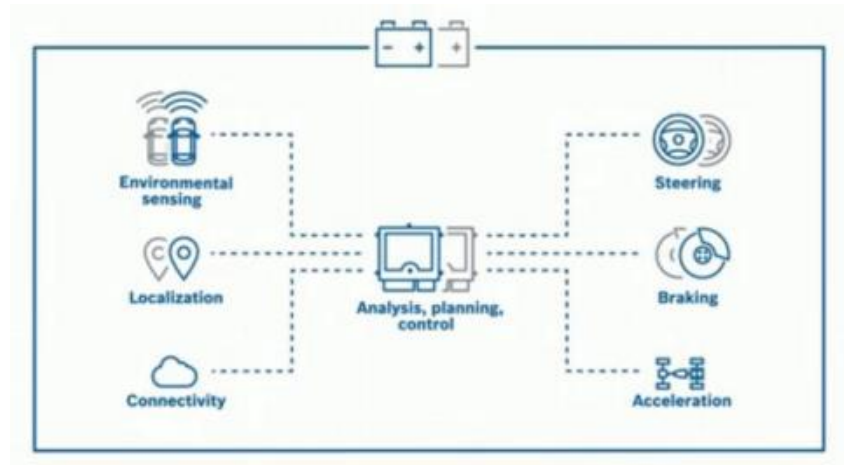


Figura 7 Sistemas redundantes de un vehículo autónomo

Un ejemplo real de un sistema redundante es el sistema de frenado que propone la empresa Bosch, donde los frenos tienen dos actuadores independientes entre sí que pueden desacelerar el vehículo esté o no pulsando el freno el conductor. Cualquiera de estos dos actuadores puede evitar el bloqueo de las ruedas al modular la presión del freno, lo que permite mantener la capacidad de control durante la desaceleración.

Todos estos sistemas redundantes deben tener diferentes fuentes de potencia y alimentación, así como diferentes redes de comunicación, para garantizar que los sistemas no fallen independientemente del lugar donde se produzca el fallo.

2.2.4 IoT y los vehículos autónomos

Se puede describir brevemente el concepto de internet de las cosas o IoT, de sus siglas en inglés, como la conectividad de múltiples dispositivos a través de internet, estos pueden interactuar entre sí sin requerir la interacción de una persona. Los objetos ya no se relacionan solo con el usuario, sino que ahora están conectados a los objetos circundantes y a bases de datos. IoT describe un mundo donde casi cualquier cosa puede conectarse y comunicarse de forma inteligente en tiempo real.

Su avance ha sido posible gracias al desarrollo de nuevas tecnologías en sensores, sistemas integrados y sistemas de análisis en tiempo real y de aprendizaje automático, pero en especial gracias a las nuevas tecnologías de telecomunicaciones y al avance continuo de las redes LTE, que conducen a redes 5G avanzadas. Con ellas se pueden proporcionar capacidades de respuesta con una baja latencia y gran ancho de banda que propician una comunicación de grandes cantidades de datos en tiempo real.

Los vehículos autónomos con Inteligencia Artificial (IA) son una variante de un sistema IoT y estos contendrán a su vez diversos sistemas IoT en su interior, será necesaria una estrecha colaboración de los fabricantes de ambos tipos de sistemas. Dentro de un vehículo autónomo se pueden dar, por tanto, tres casos diferentes de conexión entre los sistemas.

- Que los dispositivos IoT se puedan conectar vía Wifi o mediante otra tecnología a la inteligencia artificial del vehículo.
- Que los dispositivos IoT del vehículo puedan conectarse a otros dispositivos IoT que también se encuentren dentro del vehículo.
- Que los dispositivos IoT se comuniquen con otros dispositivos IoT que se encuentren fuera del propio vehículo.

Surge a raíz de este concepto el problema conocido como “factor de pantano” (*swamp factor*) según el cual el vehículo autónomo podría recibir demasiada información tanto del interior como del exterior, lo que podría provocar un fallo en la conducción, ya que los recursos disponibles dentro del vehículo son limitados. La cantidad de datos que pueden recopilarse solo del propio vehículo es muy alta, su ubicación, velocidad, cuánto tiempo lleva en carretera, consumo de combustible, entre otras muchas. Toda esta información se puede transmitir en tiempo real al propietario del vehículo.

El propio procesador del vehículo sería quien debería gestionar de alguna forma toda esta gran cantidad de información, implementando algún filtrado que separe la información útil de la irrelevante, la cual debe ser aislada.

Llegados a este punto es fácil entender cómo se pueden interconectar el concepto de IoT con los vehículos autónomos. Además de los sensores principales ya presentados, un vehículo autónomo incluye otra gran variedad de sensores que le permiten realizar esta conducción autónoma, en el siguiente apartado se verán en detalle algunas de estas tecnologías.

2.2.5 Fusión de sensores e Inteligencia Artificial

Los sensores autónomos permiten que el vehículo supervise su entorno, detecte obstáculos y planifique su ruta. En combinación con el software automotriz y las unidades de procesamiento se busca conseguir una automatización completa en el proceso de la conducción. Como su propio nombre indica, este proceso toma toda la información de los sensores, la procesa mediante técnicas de inteligencia artificial (IA), basadas en unas redes de clasificación de imágenes, que le permiten identificar que son los objetos que está viendo a su alrededor.

Los vehículos autónomos emplean los datos obtenidos por los sensores, junto con los mapas generados por el sistema LIDAR, trabajando en conjunto con las cámaras y el radar.

Empleando todo este conocimiento el sistema de Inteligencia Artificial incorporado en el vehículo consigue situarse este de forma exacta en el mapa y en su entorno, pudiendo tomar decisiones sobre el manejo y guiado de la forma más óptima y segura posible. Este tema se tratará en profundidad más en el siguiente apartado.

Para realizar estas grandes labores de cómputo se emplean procesadores de alto rendimiento. Un ejemplo es la NVIDIA DRIVE AGX. Esta plataforma de cálculo está expresamente diseñada para la conducción autónoma, incorpora hasta seis tipos diferentes de procesadores que le

permiten ejecutar los algoritmos para el procesado de los datos de los sensores, cartografía digital y capacidades de conducción, todo en tiempo real.

Si bien los sensores y sistemas empleados para la navegación y detección del entorno son los que más atención captan, hay que tener en cuenta que una gran parte del diseño recae en la administración de la energía, pues cada sistema tiene unas especificaciones diferentes, que involucran diferentes márgenes de potencia, consumo y disipación térmica.

Además, hay empresas que se dedican a la creación de mapas cartográficos de alta definición que son incorporados en el software de los vehículos, para lograr un posicionamiento con mayor precisión y permitir una conducción autónoma por esos tramos ya cartografiados, aunque el vehículo no disponga de más sistemas de visión del entorno, por ejemplo, un vehículo que no disponga de un sistema LIDAR incorporado.

Esta imagen siguiente muestra una zona ya cartografiada donde se han indicado y diferenciado todos los carriles, los límites de los mismos y las direcciones que puede tomar un vehículo indicadas mediante líneas con flechas.

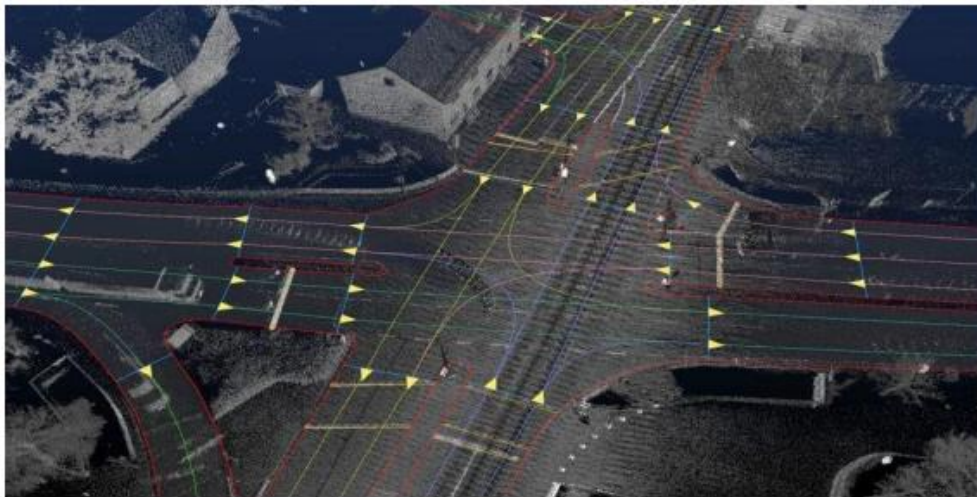


Figura 8 Mapa cartografiado para conducción autónoma

Este tipo de mapas complementan a los sistemas de conducción autónoma completa, pero no hacen posible dicha conducción solo con estos mapas, ya que el vehículo no podría identificar otros vehículos de forma precisa, están más bien pensado como un apoyo en caso de un sistema fallase o para vehículos de niveles de autonomía inferiores al 5, donde el conductor humano tiene que estar pendiente del recorrido, pero puede dejar el coche en modo autónomo.

2.3 Inteligencia Artificial y Machine Learning

La Inteligencia Artificial es la inteligencia llevada a cabo por maquinas, es la capacidad de un sistema para interpretar correctamente los datos externos, para aprender de dichos datos y emplear esos conocimientos para maximizar sus posibilidades de éxito y eficiencia en una tarea u objetivo propuesto a través de una adaptación flexible. Coloquialmente el termino inteligencia artificial se aplica cuando una maquina imita las funciones cognitivas que los humanos asocian al cerebro, como el razonamiento lógico. Dentro del campo de la conducción autónoma la inteligencia artificial juega un papel fundamental, ya que se podría considerar que es el cerebro del mismo.

El área de la inteligencia artificial es muy amplia y abarca muchas áreas muy diversas relacionadas entre sí, de entre ellas se va a destacar por su implicación en el proyecto el *machine learning*, generalmente traducido como aprendizaje automático. Este se podría definir como el campo de la inteligencia artificial que proporciona a los ordenadores la capacidad de aprender sin ser explícitamente programados, no necesitan que el programador fije una serie de reglas que debe seguir para lograr su objetivo, sino que las hace de forma automática.

2.3.1 Redes Neuronales Artificiales

Un caso particular dentro de los algoritmos de *machine learning* son las Redes Neuronales Artificiales o ANN de sus siglas en inglés (*Artificial Neural Networks*). Estas redes tienen un funcionamiento inspirado en las neuronas humanas. Las ANN consisten en un conjunto de pequeños sistemas conectados entre sí que transmiten señales, la información de entrada atraviesa la red neuronal, donde se somete a diversas operaciones, produciendo unos valores de salida.

La finalidad de estas redes es resolver problemas de forma individual o combinadas con otros métodos para realizar tareas de clasificación, identificación, diagnóstico, optimización o predicción.

Las neuronas son la unidad básica de procesamiento dentro de una red neuronal, cada una de ellas tendrá una serie de estímulos o valores de entrada a los cuales aplicará una operación, que por lo general suele ser una suma ponderada de sus valores de entrada a las que se aplica una función de activación para conseguir una señal como valor de salida que a su vez puede ser la entrada de otra neurona, generando así una red neural.

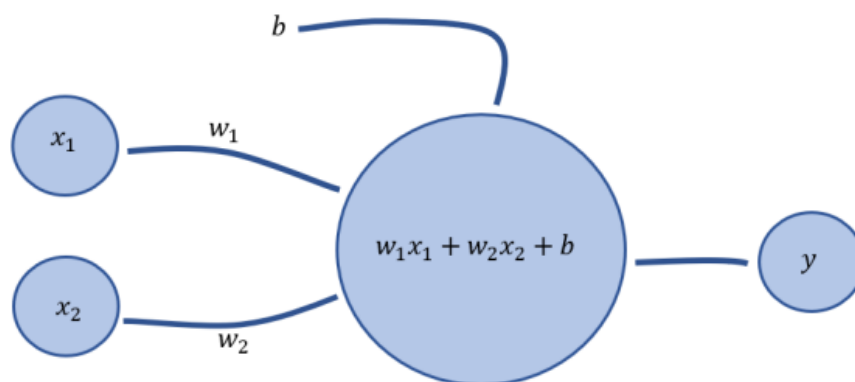


Figura 9 ANN: Neurona

El modelo se compondrá del vector " x_i " que representa los valores de entrada a la neurona. Un vector de pesos " w_i " de un tamaño equivalente al número de conexiones de entrada, donde el valor particular de cada elemento indica el nivel de intensidad con el que dicha variable de entrada afecta a la neurona, además de añadir un sesgo " b " a la operación. Los valores de los pesos no son fijos, pueden modificarse para adaptarse a la función requerida mejorando así el resultado final. La operación que se realiza dentro de una neurona es una suma ponderada de los valores de entrada por los pesos asignando a cada uno, el resultado será el valor de salida " y ".

Las redes neuronales se organizan en tres grupos o "capas". Los datos entraran por la *capa de entrada* (input), que pasan a través de la *capa oculta* (layer1, layer2, ...) formada a su vez por una cantidad variable de capas o layers, que realizan el trabajo especificado para la red. Los resultados salen por la *capa de salida* (output). Cuantas más capas tenga la red, más complejo podrá ser conocimiento que se elabore, esta profundidad en la cantidad de capas es lo que da nombre al aprendizaje profundo o *deep learning*.

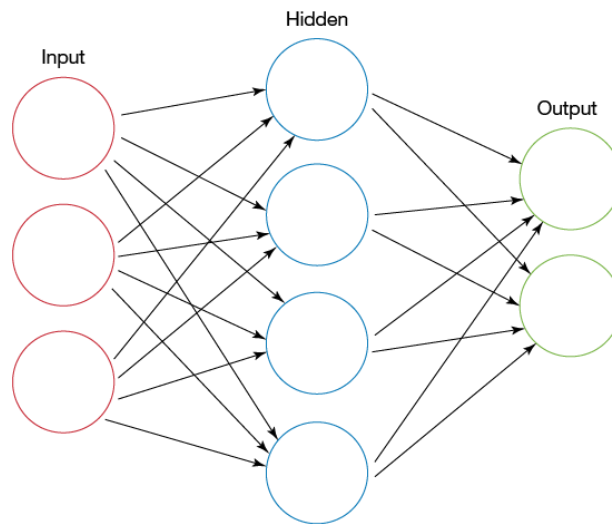


Figura 10 ANN: Estructura

La conexión de varias neuronas de forma secuencial concatenando sus resultados con las entradas de las siguientes neuronas no basta para que la red de una respuesta valida, ya que es necesario aplicar una transformación no-lineal al resultado de las operaciones que realiza cada neurona internamente. Esto se debe a que realizar la suma de todos los resultados de cada una de las neuronas equivale a solo haber realizado una única operación. Las funciones de activación son las encargadas de realizar esta modificación.

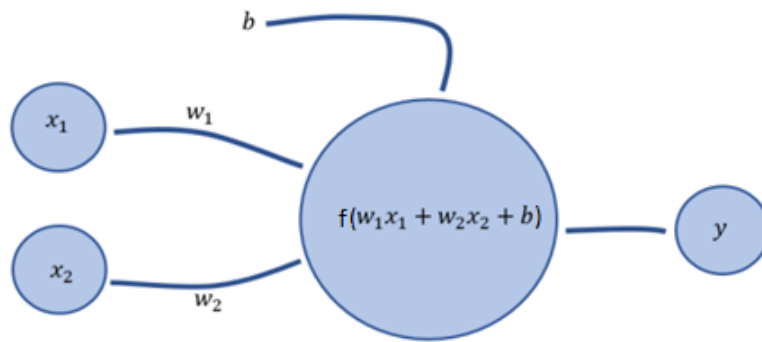


Figura 11 ANN: Neurona con función de activación

2.3.1.1 Funciones de activación

El efecto no-lineal aplicado depende de la función de activación que se implemente, la más simple es la función umbral o función escalón. Esta función indica que, si el resultado es cero, la neurona va a ser cero, pero si el resultado es mayor que cero, la salida será uno.

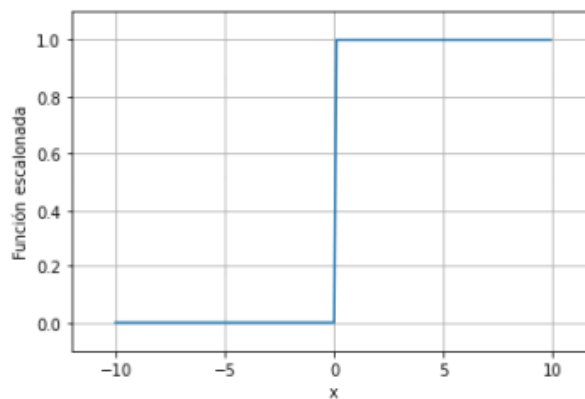


Figura 12 Función de activación escalón

El cambio instantáneo entre 0 y 1 dificulta el aprendizaje de la red, por este motivo esta función no es muy utilizada.

$$f(x) = \begin{cases} 0, & \text{para } x < 0 \\ 1, & \text{para } x \geq 0 \end{cases}$$

Ecuación 1 Función Escalón

La función de activación sigmoide, también conocida como función logística es una versión mejorada de la función escalón, esta hace que el resultado la suma ponderada sature los valores muy grandes, haciendo que valgan 1 y los muy pequeños saturen en 0, es muy utilizada para representar probabilidades. Su expresión matemática, así como su representación gráfica son las siguientes:

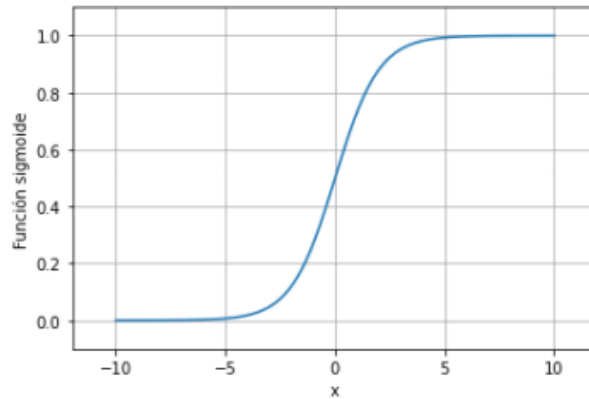


Figura 13 Función de activación sigmoide

Siendo su ecuación:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Ecuación 2 Función Sigmoide

Similar a la función sigmoide esta la función tangente hiperbólica, esta tiene un rango comprendido entre -1 y 1. Esta se puede ver en la siguiente figura junto con su expresión matemática.

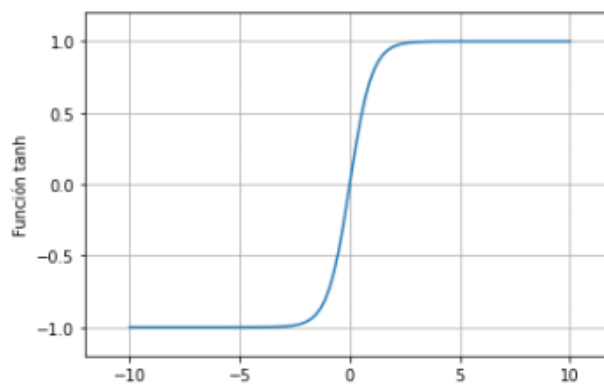


Figura 14 Función de activación tanh

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Ecuación 3 Función tanh

Finalmente, la función de activación de unidad rectificada lineal (abreviada como ReLU por sus siglas en inglés) se comporta como una función lineal cuando es positiva y como una constante a cero cuando el valor es negativo. Actualmente esta función y algunos derivados suyos son los que más emplean las redes actuales debido a que permiten un aprendizaje muy rápido.

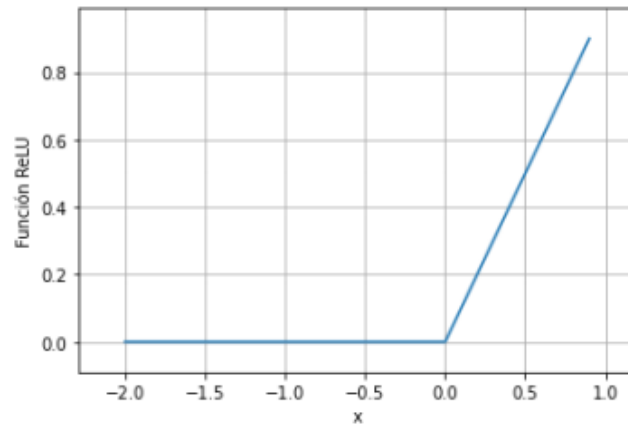


Figura 15 Función de activación ReLU

$$f(x) = \max(0, x) = \begin{cases} 0, & \text{para } x < 0 \\ x, & \text{para } x \geq 0 \end{cases}$$

Ecuación 4 Función ReLU

2.3.1.2 Backpropagation

Todos los elementos enumerados hasta este punto permiten formar una red neuronal, en la que, variando los pesos asignados a cada entrada, pueda modelar información compleja, pero se requiere de un individuo que modifique estos valores para que la red pueda realizar la tarea para la que se ha creado. Aquí entra en juego la inteligencia artificial y el *machine learning*, evitando que el usuario tenga que modificar los pesos y sea la propia red la encargada de hacerlo por su cuenta. El algoritmo que realiza esta labor se conoce como *backpropagation*.

Este algoritmo emplea el descenso del gradiente como base para realizar los cambios en los pesos. El descenso del gradiente consiste en evaluar el error del modelo en un punto y calcular sus derivadas parciales en ese punto, esto genera un vector de direcciones que indica la pendiente de la función hacia donde el error se incrementa (el gradiente), de forma se puede mover el punto en la dirección contraria. Aplicando esto de forma iterativa se puede reducir el error progresivamente.

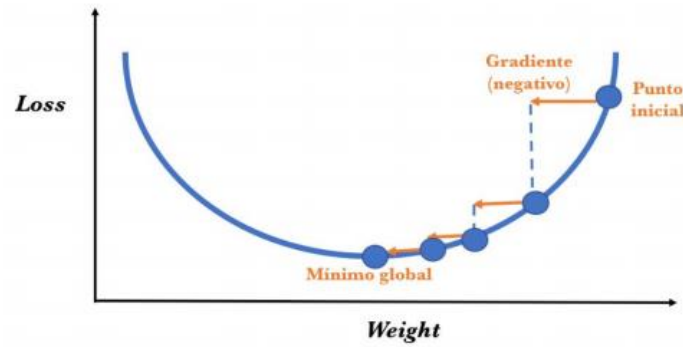


Figura 16 Aplicación del descenso del gradiente

En el caso de las redes neuronales aplicar este método es más complejo ya que modificar un parámetro dentro de la red puede cambiar en gran medida el resultado, dado que este influye en las salidas parciales de multitud de neuronas. El algoritmo de backpropagation se aplica en dos fases, en la primera los valores de entrada pasan por toda la red neuronal generando un valor de salida, como se explicó anteriormente. La segunda fase consiste en propagar el error generado por esos valores iniciales hacia atrás, hacia cada una de las capas ocultas, de esta forma se reparte la señal de error en función de la influencia o peso que tenga cada neurona dentro de la red.

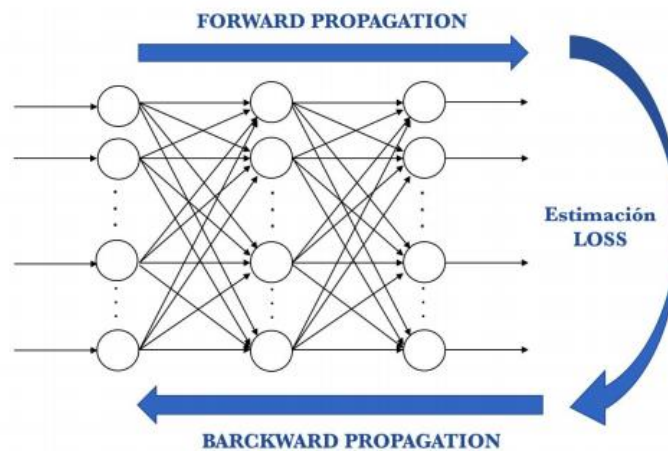


Figura 17 Backpropagation en una red neuronal

Empleando este sistema la red puede ver el error que genera cada una de sus neuronas de forma individual y así modificar el peso de la misma para reducir el error empleando el descenso del gradiente

2.3.2 Paradigmas de Aprendizaje

El punto de partida para entender como un algoritmo de aprendizaje autónomo trabaja es saber que paradigma de aprendizaje implementa, es decir, como usa los datos de entrada para aprender.

Estos mecanismos permiten procesar toda la información nueva que recibe como parámetros de entrada la red para transformarla en conocimiento. Entender los tipos de procesos es algo fundamental dentro del campo del *machine learning*, ya que todas las técnicas y algoritmos se pueden clasificar en los tres grandes grupos que se explican a continuación, en función del paradigma de aprendizaje que estén aplicando, para los cuales no es necesario saber cómo realizan dicho aprendizaje, solo entender como interactúa el sistema de inteligencia artificial con los datos de entrada y de salida.

2.3.2.1 Aprendizaje supervisado

Este tipo de aprendizaje busca descubrir las relaciones existentes entre unas variables de entrada y unas variables de salida enseñando a los algoritmos cual es el resultado que se quiere obtener para un determinado valor. Tras mostrar muchos ejemplos durante una fase conocida como entrenamiento de la red neuronal, el algoritmo será capaz de dar un resultado correcto incluso cuando se le muestren valores que no ha visto nunca aplicando las reglas que ha aprendido.

Se basan en la aplicación del algoritmo de backpropagation, ya que deben tener una salida conocida durante el entrenamiento para poder realizar las correcciones en los pesos, de modo que las salidas predichas estén más cerca de las salidas conocidas.

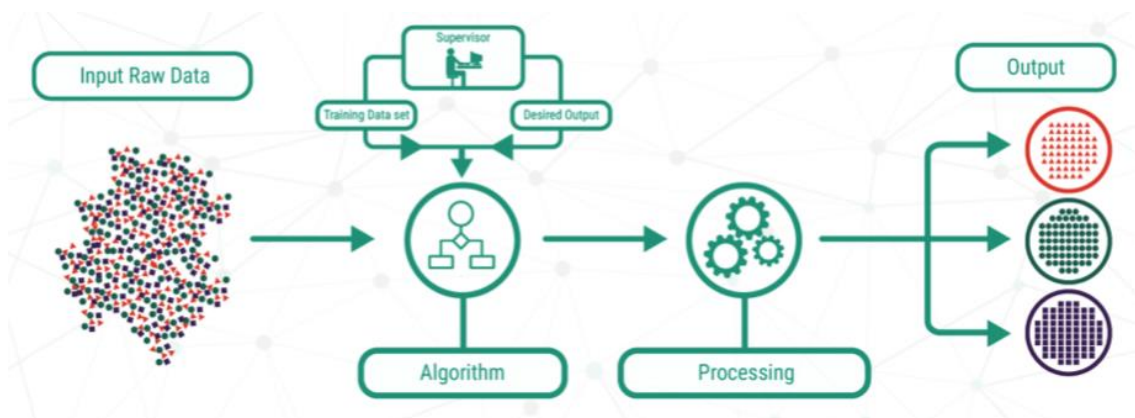


Figura 18 Aprendizaje supervisado

Es supervisado por que al mostrarle los resultados que queremos conseguir al algoritmo estamos participando en la supervisión de su aprendizaje. Este ha sido el paradigma de aprendizaje que más aplicaciones prácticas ha tenido en la última década, liderando la nueva alza que ha vivido la inteligencia artificial.

Un ejemplo de un aprendizaje supervisado podría ser el tomar datos de conducción real capturados por los sensores implementados en un vehículo y hacer que sea el algoritmo el que aprenda por imitación cuales son las acciones realizadas por el humano, la limitación en esta estrategia está en el volumen y variedad de los datos recopilados.

La mayor desventaja de este método de aprendizaje es la gran cantidad de datos que requiere, puesto que estos deben estar clasificados o etiquetados de forma manual por un ser humano antes de pasarlos al sistema de aprendizaje.

2.3.2.2 Aprendizaje no supervisado

Por definición, el aprendizaje no supervisado es el paradigma que consigue producir conocimiento únicamente de los datos que se proporcionan como entrada, sin necesidad en ningún momento de explicarle al sistema que se quiere obtener.

Este paradigma busca patrones de similitud entre los datos de entrada, pudiendo encontrar a la perfección la estructura interna que han generado dichos datos. Pueden saber si una cosa es similar a otra donde un humano no podría debido a limitaciones analíticas o sesgos durante el análisis. El problema de estos algoritmos es que no se tiene ningún ejemplo de respuesta para saber si el sistema está funcionando de forma correcta.

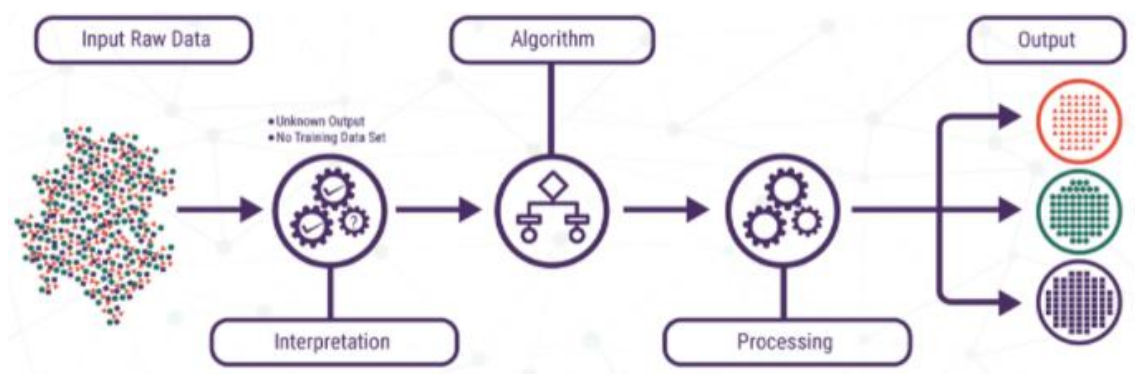


Figura 19 Aprendizaje no supervisado

Un ejemplo donde se emplean este tipo de paradigmas es en el campo de la ciberseguridad, donde el método de ataque siempre cambia y no se sabe exactamente que se está buscando. Como el algoritmo no busca una etiqueta específica, sino cualquier patrón que este fuera de la norma, el algoritmo crea una base en cómo debe verse el flujo de datos de forma normal. Si, por ejemplo, alguna transferencia rompe este patrón de comportamiento normal, esos datos se marcarían como peligrosos. Un programa basado en el aprendizaje supervisado solo podría detectar ataques que ya ha visto antes, haciéndolo ineficiente en entornos que cambian a menudo.

La ventaja de emplear este tipo de aprendizaje reside en que obtener los conjuntos de datos para entrenar son menos costosos de conseguir, ya que no hace falta etiquetar previamente todos los datos uno por uno. Dado el gran tamaño que tienen las bases de datos hoy en día para los sistemas de inteligencia artificial, ahorrar una gran cantidad de tiempo y recursos es un factor a tener muy en cuenta a la hora de decidir como diseñar un sistema de aprendizaje autónomo. El aprendizaje no supervisado presenta un futuro muy prometedor en el campo de la inteligencia artificial.

2.3.2.3 Aprendizaje reforzado

Este tipo de aprendizaje permite que el sistema descubra los mejores valores posibles para la salida mediante un sistema de “premios” positivos y negativos. El objetivo es maximizar las recompensas. La ausencia de etiquetas completamente fijas para los datos o acciones definidas que realizar para cada situación, hacen que este paradigma pueda descubrir estrategias que ni siquiera se le han podido ocurrir a los diseñadores del mismo.

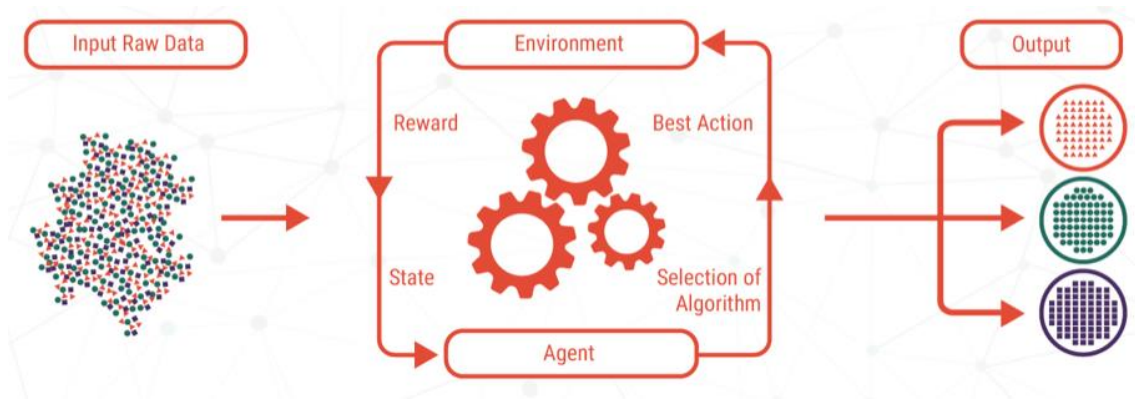


Figura 20 Aprendizaje reforzado

Un ejemplo de este paradigma aplicado a la conducción autónoma podría ser el siguiente caso. Partiendo de un vehículo autónomo, se quiere que este llegue a su destino evitando una serie de obstáculos situados por todo el camino, el vehículo sería el “agente” y la carretera el “entorno”. Cada vez que el vehículo golpee uno de los obstáculos, este habrá cometido una acción errónea y recibirá un premio negativo, de esta forma se busca que el programa entienda que no puede ir en esa dirección. En el siguiente intento el vehículo cambiará su dirección cuando llegue al obstáculo y recibirá una recompensa positiva. De esta forma el sistema aprenderá a conducir por la carretera evitando los obstáculos a base de ensayo y error.

2.3.3 Redes Neuronales Convolucionales

Una red neuronal convolucional o CNN (*Convolutional Neural Network*) es un tipo de red neuronal artificial cuyo propósito es extraer todas las características posibles de una imagen y luego emplear dichas características para detectar o clasificar los objetos de la imagen. Estas redes aplican un paradigma de aprendizaje supervisado, por tanto, es necesario proveerlas de muestras etiquetadas por un ser humano para complementar su fase de aprendizaje. Conceptualmente las neuronas de este tipo de redes serían los campos receptivos de misma forma que las neuronas de la corteza visual primaria de un cerebro biológico. Estas redes permiten a las máquinas percibir el mundo que les rodea.

Fue Yann LeCun quien en 1989 realizó el primer diseño de una red neuronal convolucional para detectar números escritos en cheques bancarios, desde entonces y dada la gran efectividad de estas redes en tareas de visión artificial, como en la clasificación y segmentación de imágenes, se ha producido un gran desarrollo en el campo de la detección de objetos.

El diseño de una red neuronal convolucional ha sido pensado para sacar partido a la estructura espacial de las imágenes, pudiendo detectar patrones complejos en enormes bancos de

imágenes. Algunos ejemplos de trabajos realizados con estas redes son la detección de caras, análisis de radiografías de pacientes o detección peatones que se cruzan ante un vehículo autónomo, como será uno de los casos de este proyecto.

Las redes neuronales convolucionales se basan en como un ser humano ve y como el cerebro analiza esa información. Si se siguen los pasos que realiza el córtex visual, este realiza un procesamiento en cascada donde primero se identifican los elementos básicos y generales y luego se centra en detalles concretos y de tal forma funciona una red neuronal convolucional.

En una red neuronal multicapa convencional al analizar una imagen, se recibirían como parámetros de entrada todos los valores de cada uno de los píxeles de la imagen en forma de vector plano, como si fueran variables independientes sin dar importancia a la posición que estos tienen dentro del conjunto completo de la imagen. Pero en una imagen, la posición de estos píxeles tiene una importancia muy relevante, el valor de un píxel está muy ligado al de los píxeles vecinos, tanto en el ancho como en el alto, generando estructuras, formas y patrones que analizadas correctamente pueden ayudar al sistema a comprender que está viendo. Si la imagen está en color tendrá también una profundidad, ya que hay tres canales de colores (RGB).

Una red neuronal convolucional se caracteriza por aplicar una capa donde se realiza una operación de convolución al dato de entrada, de ahí su nombre. Al aplicar una convolución sobre una imagen se modifica el valor de los píxeles, concretamente cada píxel nuevo que se vaya a calcular se hará colocando una matriz de números, conocida como filtro o *kernel*, sobre la imagen original y se multiplicaran y sumaran los valores de cada píxel vecino para obtener el nuevo valor del píxel.

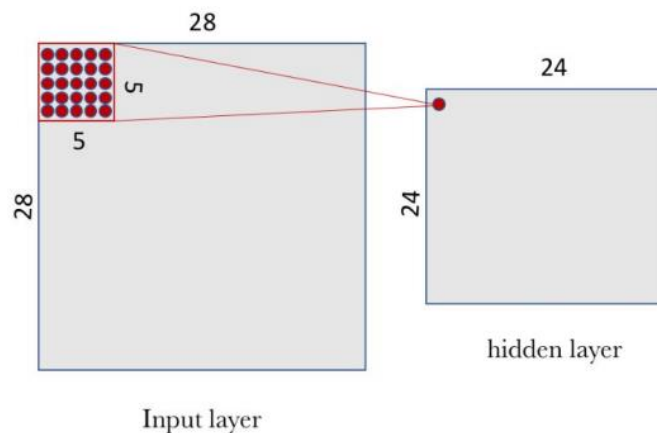


Figura 21 Aplicación del filtro en una CNN

Si esta operación se repite desplazando el filtro a lo largo de toda la imagen, como resultado se obtiene una modificación de la imagen original, lo que se vea en esta depende de los valores del filtro y son estos valores los que una red neuronal convolucional debe ir aprendiendo poco a poco. Después de la etapa de filtrado viene una de *pooling* donde se elimina toda la información que no es relevante y dejando solo los datos más representativos generados por el filtro, reduciendo así adicionalmente el tamaño de la imagen. La nueva imagen puede ser una versión borrosa o desenfocada de la original o una imagen donde se resaltan las líneas verticales u horizontales, entre otras.

La siguiente figura muestra un ejemplo donde se aplica una capa de convolución a una imagen y el resultado de esta se pasa por una función de activación de tipo ReLU, dando lugar a la imagen final, que además de tener una menor resolución que la original solo se denotan las líneas verticales.

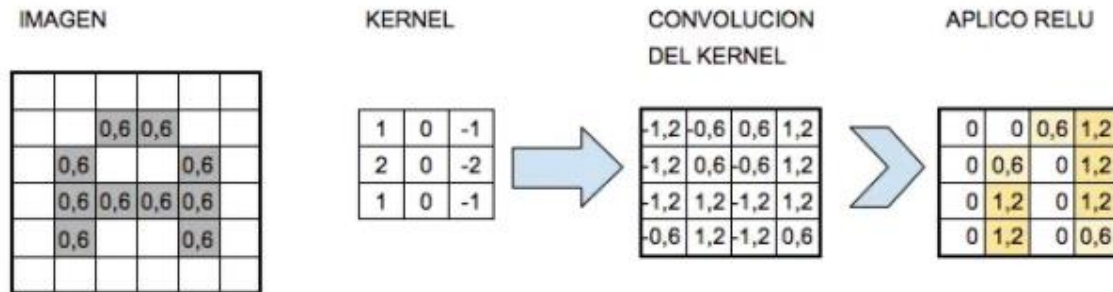


Figura 22 Aplicación de un filtro sobre una imagen

Cada una de estas nuevas imágenes generadas se le conoce como mapa de características ya que actúan como un mapa donde se indica en que parte de la imagen se ha detectado la característica buscada por el filtro. Estas imágenes generadas serán la entrada de la siguiente capa de la red.

Al volver a aplicar las operaciones de convolución sobre esta nueva imagen la propia operación se hace cada vez más potente. Por ejemplo, si el tamaño del filtro aplicado es de 3x3, donde en un inicio se tenía una región con 9 píxeles, el filtro lo convierte en un único píxel en el mapa de características generado, además, al acabar el filtrado se aplica el proceso de *polling* reduciendo el tamaño, de esta forma al pasar este mapa de características a la capa siguiente la imagen será más pequeña y menos pesada. Ya en la segunda capa se aplica ahora el filtro sobre el mapa de características volviendo a pasar de 9 píxeles a uno solo en el segundo mapa generado, lo que se está haciendo es acceder cada vez a más información espacial de la imagen original.

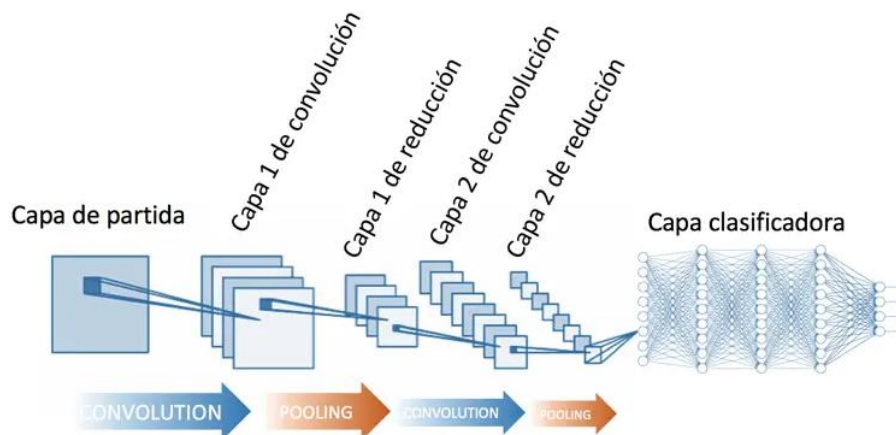


Figura 23 Estructura de una red neuronal convolucional

Una capa de convolución por sí sola no puede realizar no puede detectar formas complejas, solo líneas o curvas, pero al repetir el proceso de detección sobre el mapa de características generando por las capas anteriores permite componer cada vez patrones más complejos, como el reconocimiento de siluetas.

Debido a esta arquitectura en las redes neuronales convolucionales, la forma típica en que se representan tiene forma de embudo. La imagen inicial se va comprimiendo espacialmente, es decir, su resolución va disminuyendo al mismo tiempo que su grosor va aumentando, el número de mapas de características que se van detectado aumenta. Esto acaba cuando se tienen todos los patrones dentro de la imagen, habrá una gran cantidad de mapas de características que podrán ser los valores de entrada independientes de una red neuronal multicapa encargada de tomar la decisión sobre qué es lo que hay en la imagen.

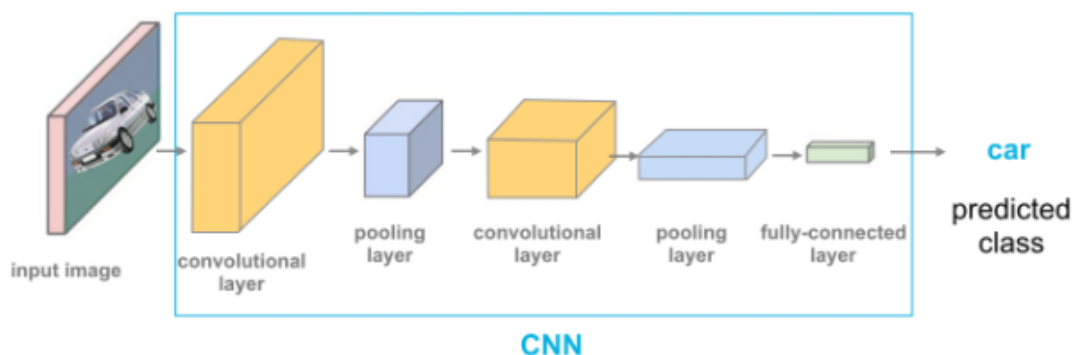


Figura 24 Arquitectura de una red neuronal convolucional

2.3.4 Redes Neuronales de Regresión

Los modelos de redes neuronales pueden ser aplicados en el análisis estadístico clásico para resolver problemas de predicción y clasificación. Lo que caracteriza a las distintas clases de modelos de regresión es la naturaleza de la variable dependiente; así, con variables continuas son modelos de regresión lineal y con variables discretas son modelos de regresión logística. Estas redes, al igual que las redes neuronales convolucionales, siguen el paradigma de aprendizaje supervisado.

Generalmente cuando se quiere poner una variable en función de otra (o de otras), se emplea el recurso de la regresión lineal (sea simple o múltiple), esta función utiliza normalmente el método de mínimos cuadrados y funciona fluidamente desde el punto de vista aritmético. El problema de esta aproximación es que, al tratarse de una recta, para los valores extremos del predictor, se obtienen valores menores que 0 o mayores que 1, lo que entra en contradicción con el hecho de que las probabilidades siempre están dentro del rango $[0,1]$.

Se emplea para este tipo de casos la regresión logística, que es un tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica (una variable que puede adoptar un número limitado de categorías) en función de las variables independientes o predictoras.

El análisis de regresión logística se enmarca en el conjunto de Modelos Lineales Generalizados (GLM por sus siglas en inglés) que usa como función de enlace la función *logit*. Las probabilidades que describen el posible resultado de un único ensayo se modelan utilizando una función logística. Una de las principales aplicaciones de un modelo de regresión logística es clasificar la variable cualitativa en función de valor que tome el predictor, en lugar de predecir exactamente 0 o 1, la regresión logística genera una probabilidad, con un valor *entre* 0 y 1, exclusivo.

Podríamos decir que los algoritmos de regresión modelan la relación entre distintas variables de entrada (*features*) utilizando una medida de error (*loss*) que se intentará minimizar en un proceso iterativo para poder realizar predicciones “lo más acertadas posibles”. Para conseguir esta clasificación, es necesario establecer un *threshold* de probabilidad a partir de la cual se considera que la variable pertenece a uno de los niveles.

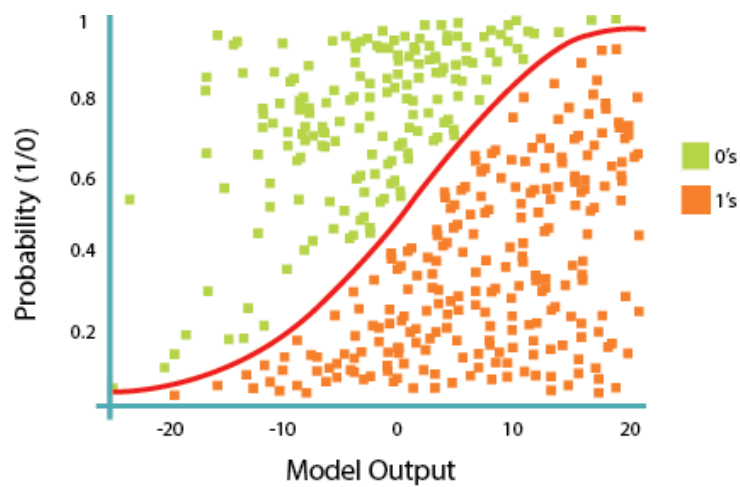


Figura 25 Gráfica de una Regresión Logística

3 Implementación

Este proyecto busca crear un software basado en redes neuronales que permita a un coche comprender su entorno para que este pueda funcionar de forma autónoma, para ello el coche analizará las imágenes que capta a través de una cámara colocada en el salpicadero y decidirá si puede continuar su trayecto o debe frenar porque hay un obstáculo en su camino. En este caso los obstáculos que la red reconocerá como tal son otros vehículos, peatones y pasos de cebra. El vehículo deberá frenar si está muy cerca de alguno de estos tres.

A lo largo de este apartado se detallará toda la implementación seguida, dividiéndola en cuatro partes diferenciadas, cada una de ellas es un punto clave dentro del desarrollo de cualquier red neuronal. Previamente al desarrollo de esta implementación ha sido necesario estudiar y aprender el lenguaje de programación Python ya que es uno de los más empleados dentro del mundo del aprendizaje autónomo y la gran mayoría de repositorios se encuentran en este lenguaje.

Adicionalmente ha sido necesario realizar una investigación sobre toda la teoría mencionada en el apartado anterior para conseguir una base en los principios básicos sobre el funcionamiento y desarrollo de redes neuronales y sistemas de aprendizaje autónomo y de esta forma comprender como funcionan las dos redes implementadas.

Este apartado esta subdividido a su vez en cuatro partes diferenciadas, en primer lugar, se trata que es y como se ha creado el banco de imágenes o *dataset* de entrenamiento para la primera red, a continuación, se explicará el funcionamiento de la primera red del sistema, una red neuronal convolucional y sus características. De igual manera se tratará la segunda red del sistema, una red neuronal de regresión.

La unión de las dos redes dentro de un mismo sistema se detalla en el último punto, para posteriormente presentar los resultados conseguidos en el siguiente apartado.

3.1 Creación del dataset

Por definición, un conjunto de datos o *dataset* es una colección de datos tabulada, donde cada columna de la tabla representa una variable en particular y cada fila representa a un miembro determinado del *dataset* que se está analizando. Cada uno de esos valores se conoce por el nombre de dato y el dataset puede incluir datos para uno o más miembros en función de su número de filas.

Los algoritmos aprenden de los datos, ahí encuentran relaciones y desarrollan comprensión para tomar decisiones y evaluar su funcionamiento, todo a partir de los datos de entrada o datos de capacitación que reciben. Cuanto mejores sean los datos de entrenamiento, mejor funcionara el modelo, es más, la calidad y cantidad de estos tiene tanto que ver con el éxito del sistema como el propio algoritmo. Pero no solo se necesita una gran cantidad de datos bien estructurados, es necesario que estos estén etiquetados de forma que mejor se adapte al modelo que se quiere implementar.

La primera red del sistema se trata de una red convolucional la cual se quiere que analice las imágenes captadas por la cámara colocada en el frontal del vehículo, por tanto, el primer *dataset* debe estar compuesto por una colección de imágenes. Para crear un *dataset* acorde al proyecto, se han recopilado un total de 450 imágenes escogidas a lo largo de diferentes *datasets* publicados en GitHub, buscando que estas encajen lo mejor posible en el marco del proyecto.

Para otro proyecto basado en un sistema de reconocimiento de imágenes valdría con un *dataset* mucho menor, pero al tratarse de un sistema de conducción autónoma pasa todo lo contrario, cuanto mayor sea el *dataset*, mejor funcionara el sistema y menos errores tendrá en la detección de elementos de su entorno, debido a que un sistema de estas características trabajara en tiempo real y debe analizar una gran cantidad de casos diferentes constantemente. Ese es el motivo por el que se ha recopilado una colección de imágenes tan grande.

3.1.1 Etiquetado con labellmg

Para la creación del *dataset* se ha empleado la aplicación “*labellmg*”, esta es una herramienta de anotación grafica para imágenes, desarrollada en Python y emplea Qt como interfaz gráfica. Una característica destacable de esta aplicación es permitir guardar las imágenes etiquetadas en una gran variedad de formatos, entre ellos el formato YOLO empleado por la red neuronal convolucional implementada en este proyecto. La siguiente imagen es un ejemplo donde están etiquetados a mano todos los casos que deberá analizar la red convolucional, en este caso 4 vehículos, 2 personas y 2 pasos de cebra.

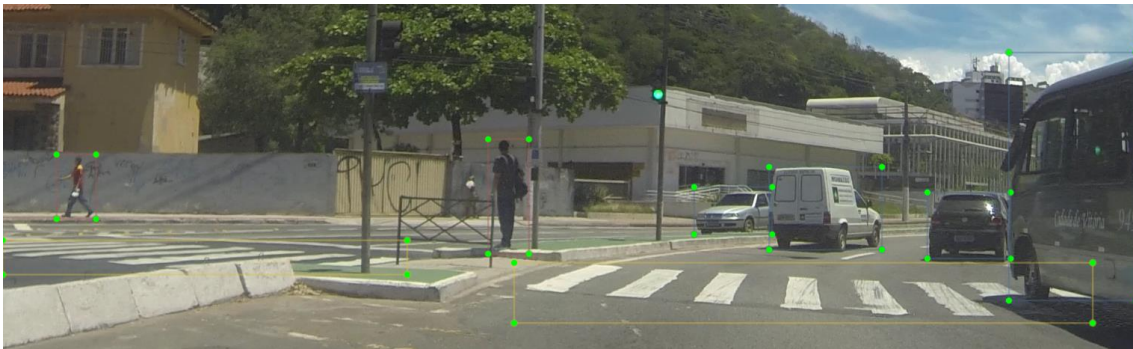


Figura 26 Imagen etiquetada con labellmg

El programa simplemente crea un archivo XML ligado a la imagen en el que se encuentran las coordenadas de cada cuadro creado en la interfaz gráfica en el formato seleccionado, en este caso, formato YOLO. Este mismo proceso se seguirá con las 450 imágenes seleccionadas para nuestro *dataset*.

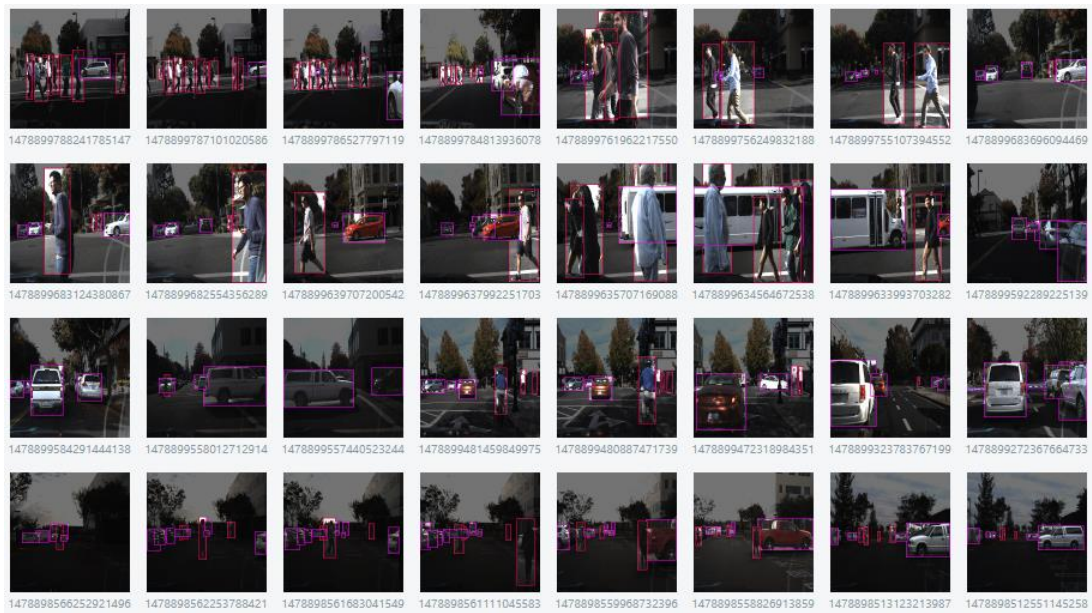


Figura 27 Dataset completo etiquetado

3.1.2 División del dataset en grupos de entrenamiento y test

Con todo el *dataset* etiquetado es necesario dividir este en dos grupos, uno para entrenamiento y otro para test. El primero lo empleara el sistema para aprender las características de cada uno de los objetos que debe detectar dentro de una imagen, para posteriormente cuando el sistema ya haya entrenado, analizar las imágenes del segundo grupo, el conjunto de test. Estas son imágenes que el programa no habrá visto nunca y analizará en base a los conocimientos adquiridos en la fase de entrenamiento. Al tratarse de imágenes también etiquetadas, se puede medir como de bien ha realizado la detección nuestro sistema comparando las etiquetas de ambos casos.

Esta división en dos grupos se suele realizar de forma aleatoria pero siempre buscando que en ambos grupos haya una buena representación de todos los datos que el sistema debe tratar. En cuanto a la cantidad de datos de entrenamiento necesario u óptimos para un modelo, no hay una regla estricta sobre la cantidad, todo depende de la complejidad del modelo creado y la cantidad de datos que este deba tratar, en el caso de la conducción autónoma se requieren una gran cantidad de datos para que el modelo sea muy seguro y no cometa a penas errores.

Para realizar esta división de forma sencilla se emplea la página “*Roboflow*”. Esta página permite evaluar la calidad de un conjunto de datos y prepara el mismo para ser exportado a nuestra red, realizando la división del conjunto completo en los dos grupos y realizando un reescalado del tamaño de las imágenes a 416x416, tamaño con el que trabaja la red convolucional.

La siguiente imagen presenta los resultados del análisis del *dataset*, donde se muestran un total de 3109 anotaciones con una media de casi 7 anotaciones por imagen. Además de enumerar cuantas anotaciones hay por cada dato.

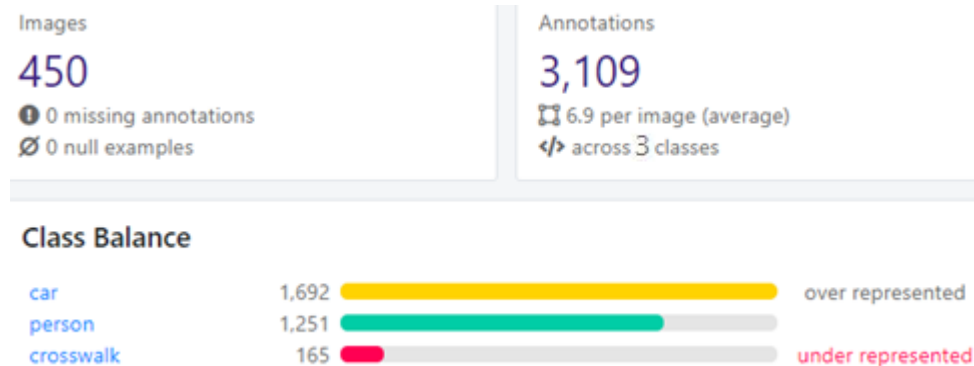


Figura 28 Evaluación del dataset

La baja cantidad de casos de pasos de cebra en el conjunto total de imágenes desemboca en una menor representación dentro del conjunto completo, pero esto es algo común dentro de los *datasets*. Por lo general no todos los datos dentro de un conjunto de imágenes estarán representados de igual forma, en este caso se ha dado prioridad a la detección de otros vehículos y peatones.

Tras finalizar el análisis del conjunto de imágenes, se aplica una división aleatoria al *dataset* para crear los dos grupos, entrenamiento y test. La página nos permite elegir qué porcentaje de imágenes se destinará a cada grupo encargándose de que en ambos grupos todos los datos estén bien representados. Por tanto, se ha destinado el 70% para el conjunto de entrenamiento y un 30% para el conjunto de test. De esta forma la creación del *dataset* está completa y se puede pasar a analizar el funcionamiento de la red convolucional.

3.2 Red neuronal convolucional: YOLOv5

Los seres humanos pueden mirar una imagen y reconocer al instante los objetos que la forman, esto les permite poder realizar tareas complejas como la conducción de un automóvil, pudiendo detectar cualquier objeto en la vía con solo un golpe de vista. Bajo esta premisa funciona la red neuronal convolucional empleada, una red ya diseñada y funcional, especializada en la detección de imágenes y de código abierto, conocida por el nombre de YOLO (*"You Only Look Once"*). El algoritmo en el que se basa una red YOLO es rápido y preciso en la detección de objetos dentro de una imagen, de esta forma se busca emular la visión humana, en este caso, dentro de un vehículo autónomo.

3.2.1 Funcionamiento de una red YOLO

Una de las principales características de una red YOLO es el poder procesar hasta 45 fotogramas por segundo, para conseguirlo la red aprende representaciones generales de los objetos que se desean detectar, haciéndola muy eficiente en la detección de objetos en segundo plano. Pero esta velocidad tiene un coste, una red de este tipo puede cometer más errores que otras redes convolucionales especializadas también en la detección de objetos.

La red ve la imagen completa tanto en las fases de entrenamiento como las de test, pudiendo codificar la información contextual de las clases y la apariencia de cada una de ellas, creando así un cuadro delimitador para cada una. Es decir, la red razona sobre la imagen al completo y los objetos que la componen.

La detección de objetos en una red YOLO comienza con la división de la imagen en una cuadrícula de $S \times S$ cuadrados, de forma que, si un objeto cae dentro de una celda, esta será la responsable de detectarlo. Luego cada una de esas celdas tratará de detectar si hay objetos dentro y predecirá una cantidad de cuadros delimitadores, es decir, predecirá donde hay un objeto y colocará un cuadrado que lo delimite. Cada uno de estos cuadros consta de cinco predicciones, las coordenadas de su centro (x, y), el alto (h) y el ancho (w). Además, asociado a cada cuadro delimitador habrá una puntuación de confianza que indica como de segura está la red de que ahí se encuentra un objeto, si, por ejemplo, no existe ningún objeto dentro de la celda la puntuación de confianza debe ser cero.



Figura 29 YOLO: Imagen dividida en $S \times S$



Figura 30 YOLO: Combinación de cuadros delimitadores

Por otro lado, cada una de las celdas predecirá también la probabilidad de cada clase condicional, es decir qué tipo de objeto se puede encontrar en cada una de las celdas.

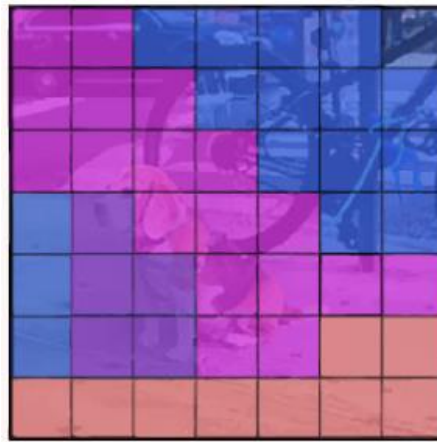


Figura 31 YOLO: Mapa de probabilidades

La mayoría de estos cuadros delimitadores tendrán una puntuación de confianza muy baja y son descartados en base a un valor umbral por debajo del cual no se considera objeto su contenido. Quedando así solo los cuadrados delimitadores que si contienen objetos. Después se multiplicarán las probabilidades de las clases condicionadas por la confianza de los cuadros delimitadores restantes, lo que da como resultado una serie de puntuaciones de confianza finales para cada casilla. Este valor final indica como de bien el cuadro estima que ahí se encuentra un objeto y si el tamaño del cuadrado delimitador está bien ajustado.

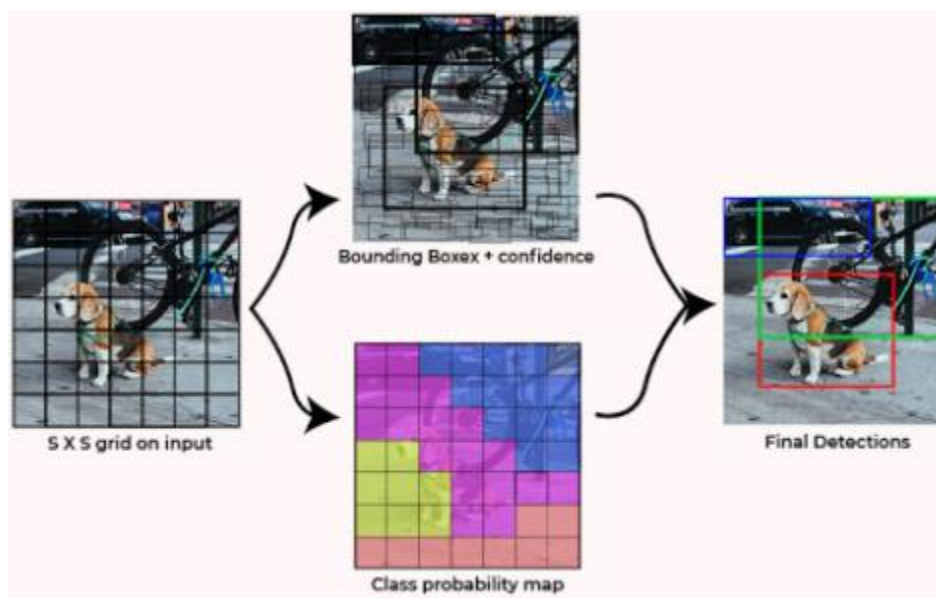


Figura 32 YOLO: Resultado final

El funcionamiento de esta red encaja a la perfección con el objetivo del proyecto y es la razón por la que se ha decidido emplear la red YOLO, pero además hay otro motivo, entrenar una red convolucional nueva desde el inicio y conseguir que esta reconozca objetos requiere de un largo periodo de entrenamiento y una gran cantidad de ajustes. Las redes convolucionales especializadas en la detección de objetos tienden a ser muy profundas con un gran número de parámetros que ajustar, además se tiene que contar con un conjunto de datos de entrenamiento muy amplio y variado, de forma que no se condicione a la red.

3.2.2 Transferencia de aprendizaje e implementación

La transferencia de aprendizaje es el método por el cual se pueden emplear redes previamente entrenadas en la detección de objetos y aprovechar sus capacidades para reconocer formas y patrones, de esta manera se puede entrenar una red para que detecte los objetos deseados sin emplear una gran cantidad de tiempo en el entrenamiento, ni requerir de un gran conjunto de datos para el mismo. En concreto la red empleada es la quinta versión de la red YOLO original (YOLOv5), publicada el 9 de junio de 2020 en formato de código abierto al igual que el resto de versiones anteriores. Esta última versión, además de ser más robusta y sencilla de usar, se caracteriza por conseguir una mayor velocidad tanto al inferir las imágenes como el entrenamiento.

Empleando “*Google Colaboratory*” como entorno de trabajo se carga en la red YOLO el *dataset* final creado en el paso previo. Este contiene dos carpetas, una con el conjunto de entrenamiento y otra con el conjunto de test, además de un archivo (“*data.yaml*”) que le indica a la red el nombre de los diferentes objetos que se van a detectar. También se tiene que fijar el número épocas de entrenamiento, que para este tipo de redes suele ser siempre mayor de 3000. En el caso de este proyecto se ha probado diferentes valores, siendo 5000 épocas la que mejor resultados ha conseguido.

3.2.3 Resultados y rendimiento

La siguiente grafica muestra la evolución de la precisión del sistema a medida que aumenta el número de épocas entrenadas, pudiendo verse en la primera mitad del entrenamiento como el sistema está aprendiendo a reconocer los objetos deseados, esto se refleja en la forma irregular de la gráfica. Para en la segunda mitad lograr una mejora progresiva en sus capacidades para identificar las tres clases. Finalmente, cuando el número de épocas es próximo a las 5000, la gráfica ya comienza a tomar una forma plana, a partir de este punto a penas mejora la precisión que se encuentra en torno a un 70%.

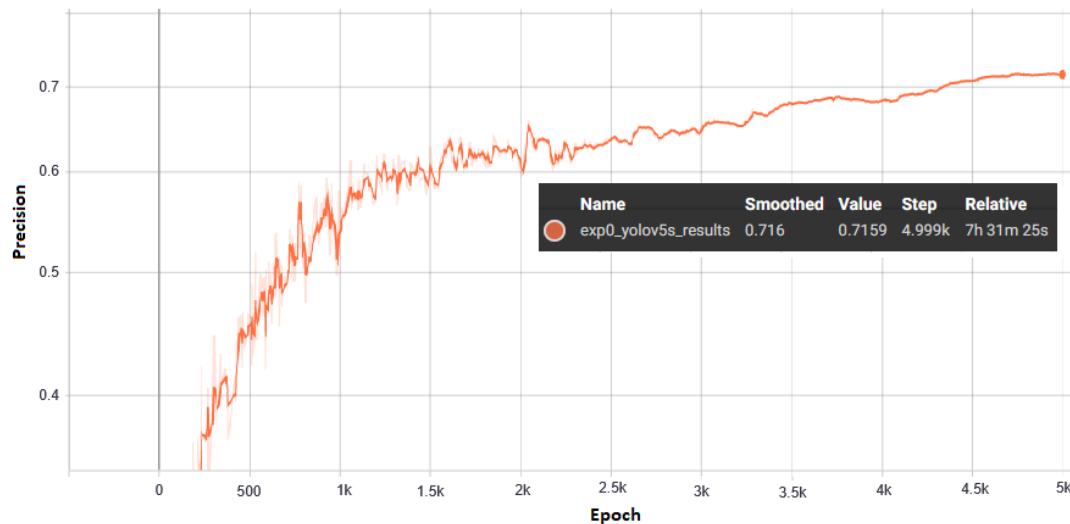


Figura 33 Resultados YOLO: Precisión vs Épocas

Al finalizar el entrenamiento se han introducido al sistema una serie de imágenes nunca vistas por el conjunto de entrenamiento ni el de test, para comprobar como reacciona ante ellas. Algunas de estas imágenes ya procesadas por el sistema son las siguientes, donde el número asociado a cada una de las etiquetas de cada objeto indica el porcentaje de seguridad que tiene la red sobre el tipo al que pertenece cada objeto detectado.



Figura 34 Resultados YOLO: Imágenes etiquetadas

Solo queda extraer el archivo generado al finalizar el entrenamiento, que contiene los pesos de todas capas que forman la red neuronal convolucional, de esta forma no será necesario volver a repetir el entrenamiento de la red cada vez que se ejecute. Gracias a esta red el sistema ya es capaz de “ver” que tiene delante, ahora es necesario que aprenda que hacer con toda esta información.

3.3 Red neuronal de regresión logística

La red neuronal de regresión logística será la encargada de generar la salida del sistema, que indicaría al vehículo si debe frenar debido a la proximidad de alguno de los obstáculos detectados por la red YOLOv5, esta decisión la tomará en función del tamaño relativo de los mismos.

3.3.1 Creación del dataset

Antes de crear la propia red de regresión, hay que crear el *dataset* para el entrenamiento y test de la red, al igual que se hizo con la red neuronal convolucional. En este caso los valores que compondrán el *dataset* serán las coordenadas que definen los cuadros delimitadores generados por la red anterior, un valor numérico que sirve de etiqueta e indique de qué tipo de objeto se trata y además un último valor binario, donde 0 indica que el vehículo puede continuar la marcha y 1 que el vehículo debe frenar.

Para obtener el tamaño de los cuadros delimitadores hay que acceder a las funciones empleadas por la red anterior, en concreto a la encargada de dibujar los cuadros delimitadores resultantes que se muestran en las imágenes finales de la red convolucional. Se añadirán una serie de líneas de código a esta función que permitirán almacenar todos los datos necesarios para la creación del *dataset* en un archivo “.csv”, facilitando así su análisis antes de introducirlos en la red siguiente. Un archivo CSV (“*comma-separated values*”) es un tipo de documento con formato abierto, muy útil para la representación de datos en forma de tabla, donde las columnas se separan por comas y las filas por saltos de línea, este formato es muy empleado para el trabajo con redes neuronales.

La función encargada de dibujar los cuadrados delimitadores trabaja con cuatro coordenadas y una etiqueta, esta última indica que tipo de qué tipo de objeto se trata y será lo primero que compruebe nuestro código, analizando cada tipo de objeto por separado. Las coordenadas empleadas para dibujar los cuadros delimitadores corresponden a los valores de los ejes X e Y del vértice superior izquierdo (“ x_0 , x_1 ”) y del vértice inferior derecho (“ x_2 , x_3 ”). La altura del objeto se obtendrá al restarle x_1 a x_3 y la anchura al restarle x_0 a x_2 . De esta forma obtendremos el tamaño relativo del objeto.

Ahora se deben fijar unos valores aproximados para los umbrales de decisión que decidirán si el vehículo debe frenar o no, en función de su altura o anchura relativa, generando así el último dato del *dataset*. Este es el dato más importante ya que gracias a él la red aprenderá como debe responder el vehículo ante cada situación y evita tener que etiquetar cada cuadro a mano. Teniendo en cuenta que esta red YOLO trabaja con imágenes de 416x416, el vehículo deberá frenar cuando:

- Un vehículo tenga una anchura mayor de 100
- Un peatón tenga una altura mayor de 130
- Un paso de cebra tenga una anchura mayor de 300

Con estas modificaciones la primera red generara al finalizar el archivo CSV con el *dataset* completo para la red neuronal de regresión. El siguiente paso consiste en introducir una serie de imágenes por la red YOLO ya entrenada para que las analice y nos genere el nuevo *dataset*.

Se han seleccionado un total de 60 imágenes pertenecientes al *dataset* empleado para entrenar la red YOLO, ya que no importa que la red las conozca, porque lo importante aquí es que marque bien los cuadrados delimitadores y los tipos de cada objeto.

Estas 60 imágenes generan un archivo CSV de 1574 líneas, cada una corresponde a los datos de un cuadrado delimitador. A continuación, se presenta un extracto de las primeras líneas de este archivo, siendo la columna final la orden que debe ejecutar el vehículo.

| | Label | x0 | x1 | x2 | x3 | Order |
|---|-------|-----|-----|-----|-----|-------|
| 0 | 2 | 122 | 249 | 130 | 277 | 0 |
| 1 | 1 | 1 | 316 | 38 | 342 | 0 |
| 2 | 0 | 147 | 253 | 158 | 275 | 0 |
| 3 | 0 | 106 | 251 | 124 | 280 | 0 |
| 4 | 0 | 132 | 253 | 149 | 278 | 0 |
| 5 | 1 | 144 | 309 | 413 | 347 | 0 |
| 6 | 0 | 374 | 248 | 415 | 280 | 0 |
| 7 | 0 | 220 | 245 | 249 | 290 | 0 |
| 8 | 0 | 71 | 250 | 102 | 287 | 0 |
| 9 | 0 | 173 | 242 | 213 | 304 | 0 |

Figura 35 Dataset de la red de regresión en formato CSV

Al igual que se hizo con el *dataset* de imágenes, es necesario dividir este en datos para entrenamiento y datos para test. Pero antes hay que indicarle al sistema que dato es el que se desea predecir, en este caso la columna "Order", para ello se separa esta columna y se almacena en una variable diferente a la del resto del conjunto de datos. La separación en los dos grupos se realiza por medio de un comando de la librería "sklearn" que permite seleccionar que porcentaje va al conjunto de test, dejando el restante para entrenamiento. En este caso se ha seleccionado un 25% de los datos para el conjunto de test, dejando el 75% restante para entrenamiento.

3.3.2 Implementación y rendimiento

También de la librería de “*sklearn*” se importará un modelo base de regresión logística que emplea un generador de números aleatorios para las asignaciones de pesos a la hora de ajustar el modelo. Esta técnica es muy utilizada en los entrenamientos de redes neuronales, pero hay que tener que cuenta que es posible que unas mismas entradas no generen los mismos datos de salida.

Tras entrenar el modelo con el *dataset* se puede analizar su desempeño a través de una serie de análisis. El primero de todos es comprobar su exactitud, siendo esta el número de predicciones correctas dividido por el número total de predicciones.

```
Accuracy: 0.9365482233502538
```

Figura 36 Exactitud de la red de regresión

El resultado es una exactitud de casi un 94%. Además, se puede calcular la precisión del sistema, entendiendo esta como la capacidad de la red para no clasificar una muestra positiva si es negativa, es decir, no etiquetar un caso donde el vehículo debe continuar la marcha como uno donde debe avanzar, ya que estas situaciones conducirían a un accidente. La precisión por tanto es la relación entre el número verdaderos positivos (*TP*) y el número de falsos positivos (*FP*).

$$Precision = \frac{TP}{(TP + FP)}$$

Ecuación 5 Formula de la precisión para una red de regresión

```
Precision: 0.7857142857142857
```

Figura 37 Precisión de la red de regresión

El valor de la precisión para este sistema es de casi un 79%. Una forma más grafica de entender el desempeño de este modelo, es por medio de una matriz de confusión (“*Confusion Matrix*”), esta solo puede usarse si se conocen los valores verdaderos de cada uno de los casos analizados por el sistema. Teniendo en cuenta que se considera un caso positivo (la salida vale 1) cuando el vehículo ha frenado de forma correcta y un caso negativo (la salida vale 0), cuando el vehículo puede continuar la marcha ya que no hay obstáculos al frente. Una matriz de confusión se divide en los siguientes cuadrantes:

- Esquina superior izquierda: **Verdadero Negativo**, casos en los que se predijo que el resultado sería negativo y el resultado real es negativo.

- Esquina superior derecha: **Falso Positivo**, casos en los que se predijo que el resultado sería positivo, pero el resultado real es negativo.
- Esquina inferior izquierda: **Falso Negativo**, casos en los que se predijo que el resultado sería negativo y el resultado real es positivo.
- Esquina inferior derecha: **Verdadero Positivo**, casos en los que se predijo que el resultado sería positivo y el resultado real es positivo.

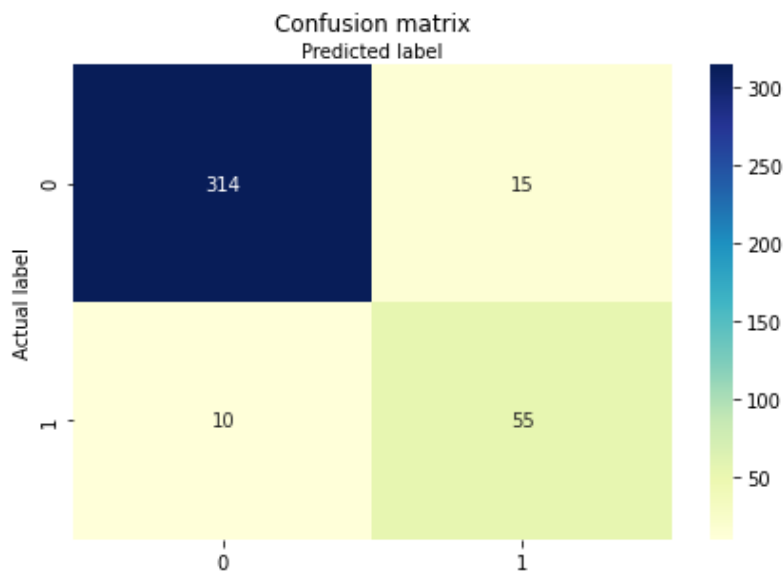


Figura 38 Matriz de confusión de la red de regresión

Finalmente, la siguiente gráfica muestra la curva de características operativas del receptor o curva ROC (*Receiver Operating Characteristic*) y representa la sensibilidad frente a la especificidad para un sistema clasificador binario según se varía el umbral de discriminación, como la proporción de verdaderos positivos varía frente a la proporción de falsos positivos. Es decir, la gráfica dice como de bueno es el modelo a la hora de distinguir entre dos cosas. La grafica está compuesta por dos líneas una de ellas es una línea recta y punteada que indica el peor caso posible, cuando el modelo no puede distinguir entre ambos casos. Un buen clasificador se encontrará lo más lejos posible de esta línea, hacia la esquina superior izquierda. La otra es la generada con los resultados de test de nuestro sistema.

El área bajo la curva ROC, también conocido como AUC (*Area under the curve*), es el valor numérico que complementa a la gráfica ROC para ver como de bien funciona el modelo. Si la puntuación AUC vale 1, representa un clasificador perfecto, mientras que 0.5 representa un clasificador sin valor, el peor caso posible. Si la AUC vale 0 significa que el modelo está prediciendo la clase negativa como positiva y viceversa.

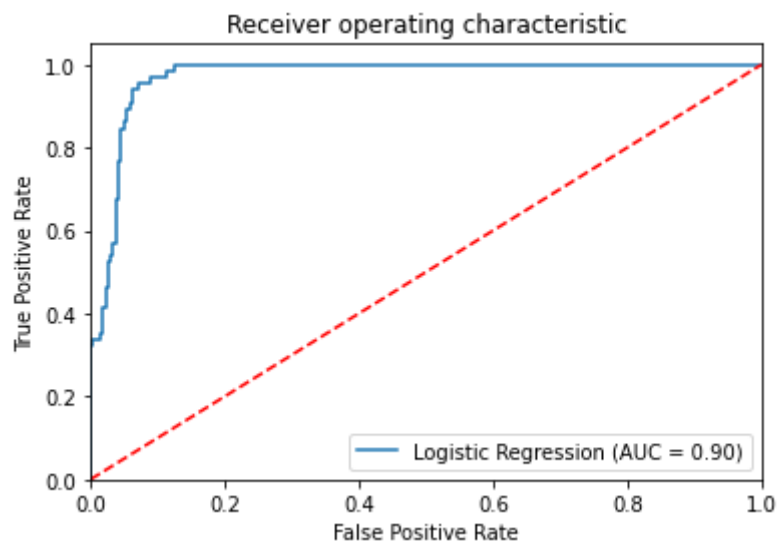


Figura 39 Curva ROC de la red de regresión

El resultado mostrado en la gráfica ROC es muy bueno, teniendo el modelo una AUC de 0.90, indicando así que el sistema logra discernir los dos posibles casos de una forma muy eficiente. Con esto finaliza el desarrollo de la segunda red del sistema, el paso final de la implementación consiste en la unión de las dos redes dentro de un mismo programa.

4 Sistema completo y resultados conjuntos

Con las dos redes funcionando de forma correcta, pero independientes, el último paso de la implementación del sistema de conducción autónoma, es juntarlas dentro de un mismo programa con la finalidad de optimizar la gestión de los datos de entrada.

La red de regresión trabaja con los valores numéricos que emplea la red YOLO para dibujar los cuadros delimitadores una vez ha detectado los objetos deseados dentro de una imagen o fotograma. Por tanto, esta segunda red ya entrenada y lista para su uso, se ha añadido dentro de la propia red YOLO.

Cuando la red convolucional YOLO se inicie a su vez se iniciará la red de regresión, esta última tomara de la función encargada de dibujar los cuadros delimitadores los tamaños relativos de los objetos finales detectados por la red YOLO y los analizara, generando como resultado una variable booleana que indica si el vehículo puede continuar la marcha (valor 0) o debe frenar (valor 1). Estas predicciones individuales para cada objeto se almacenarán en un vector, de forma que cuando todos los objetos de la imagen estén detectados y delimitados se tomara la decisión final teniendo en cuenta todas las predicciones. Con que solo una de ellas diga que el vehículo debe frenar, este lo hará. Solo siguiendo la marcha en el caso donde todas las predicciones indican que puede continuar.

4.1 Resultados finales

Con el fin de hacer los resultados finales más gráficos, se ha añadido la capacidad de generar un texto informativo a la función encargada de dibujar los cuadros delimitadores. Ahora escribirá también un texto en la parte inferior de la imagen resultado indicando la acción que debe realizar el vehículo. La siguiente imagen muestra como el vehículo equipado con la cámara frontal puede continuar la marcha en una primera instancia, pero al acercarse a otro vehículo llega un punto donde este está demasiado cerca y debe frenar. Una vez que el otro vehículo avance y este a una distancia segura, el sistema volverá a permitir al vehículo autónomo avanzar.



Figura 40 Resultados finales - Secuencia de detección

Este mismo procedimiento seguiría el sistema ante el resto de los objetos incluidos para su detección. A continuaciones se presentan una serie de capturas como muestra de las respuestas que el sistema completo ha dado ante casos donde el vehículo debe frenar debido a la proximidad de alguno de los objetos detectados.

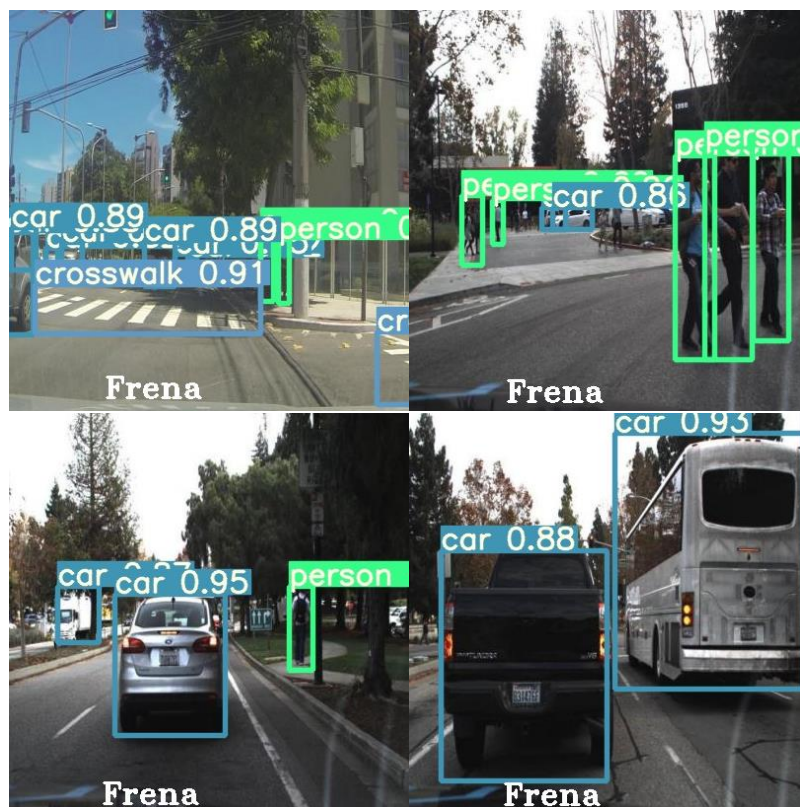


Figura 41 Resultados finales - Frena

Así como, los casos donde el vehículo puede continuar la marcha ya que los objetos detectados están aún lejos.

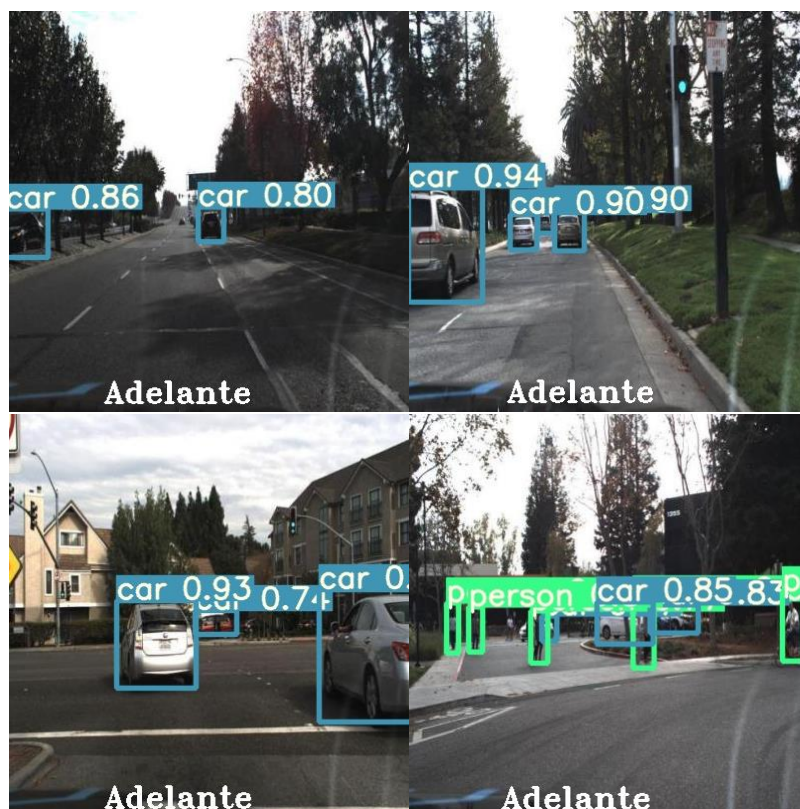


Figura 42 Resultados finales - Adelante

Adicionalmente, se pueden dar casos donde el sistema no reconoce ningún objeto en la vía, porque no hay ninguno, ante estas situaciones el vehículo podrá continuar la marcha.

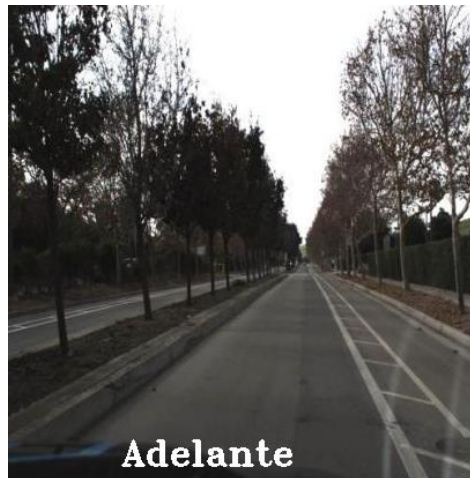


Figura 43 Resultados finales - Vía libre de objetos

Hay que señalar que el sistema no es perfecto, como ya se analizó en los resultados de rendimiento de ambas redes, los sistemas funcionan correctamente en la gran mayoría de las situaciones, pero se pueden dar cada cierto tiempo casos de detecciones erróneas que llevarían a un mal funcionamiento el vehículo autónomo. La siguiente imagen presenta uno de estos casos.



Figura 44 Resultados finales - Mala detección

Aquí el sistema YOLO ha detectado un árbol como si fuera una persona, haciendo a su vez que la red de regresión debido al tamaño de este haga frenar el vehículo. A pesar de que estas situaciones no se den frecuentemente es necesario que un vehículo autónomo no cometa este tipo de errores, ya que pone en peligro tanto a los pasajeros como al resto de personas y otros vehículos que se encuentren cerca. Las posibles soluciones a estos problemas se expondrán en detalle en el siguiente apartado.

4.1.1 Análisis de los fotogramas por segundo

Dado que el objetivo de este sistema es el poder ser implementado en un vehículo autónomo trabajando en tiempo real, es relevante realizar un estudio sobre su desempeño en este ámbito, comprobando los fotogramas por segundo que puede procesar el sistema completo. Este valor está directamente relacionado con la distancia de frenado de un vehículo autónomo, la diferencia de frenado entre un sistema que pueda trabajar a 10 FPS frente a uno que trabaje a 30 FPS, especialmente en altas velocidades, puede ser de hasta 15 metros.

Para ello se ha procesado un video de 1 min y 25 segundos de un vehículo circulando por la vía, este lleva incorporada una cámara en la misma posición que las empleadas en el *dataset*. La plataforma “*Google Colaboratory*” permite emplear una GPU para realizar la simulación del programa, en este caso la GPU empleada se trata de una Nvidia, modelo *Tesla T4*, que cuenta con 16 GB de memoria GDDR6. Las GPUs (*Graphic Processing Unit*) son coprocesadores dedicados al procesamiento de imágenes y operaciones de coma flotante y son fundamentales en este tipo de sistemas de aprendizaje autónomo, ya que acortan tanto los tiempos de entrenamiento como de simulación en gran medida.

El video analizado cuenta con un total de 2551 *frames* que han sido procesados en 80.6 segundos por la GPU. Obteniendo los siguientes resultados.

| | |
|-------------------------------------|-------|
| Tiempo por frame [s] | 0.031 |
| Fotogramas por segundo [fps] | 36.64 |

Tabla 1 Rendimiento del sistema con un video

El sistema presenta una buena respuesta en cuanto a la cantidad de imágenes que puede procesar, de querer aumentar este valor sería necesario emplear una GPU más potente.

5 Conclusión y líneas futuras

El objetivo de este proyecto ha sido aprender a un nivel básico el funcionamiento de un vehículo autónomo y complementar este estudio con el desarrollo de un software que podría ser la base de un sistema autónomo completo. Para los vehículos autónomos lo principal es la seguridad tanto del usuario como del resto de personas de su entorno y por ello emplean una gran cantidad de sensores que le permiten analizar todo lo que le rodea en gran detalle, toda esta información es recopilada y enviada a los sistemas de inteligencia artificial que se encargan de analizar los datos en conjunto para que el vehículo sea consciente de lo que le rodea, otros vehículos, personas, señales viales, obstáculos en la vía, etc. Reconocer todo lo que se encuentra alrededor del vehículo permite a este tomar decisiones durante el trascurso del recorrido en tiempo real.

En el caso particular de este proyecto se ha propuesto un sistema inteligente que analiza ciertos objetos detectados por el vehículo a través de una cámara frontal situada en el salpicadero. El sistema se encargaría de detectar a otros vehículos, peatones y pasos de cebra y con esta información decidir si el vehículo se está acercando demasiado a alguno de ellos y debería frenar, evitando de esta forma posibles colisiones o accidentes.

A lo largo del capítulo tres se han detallado todos los pasos necesarios para la creación desde cero de un sistema que cumpla esta función. Como el sistema debe analizar las imágenes captadas por la cámara, la primera red debía ser una red neuronal convolucional, ya que están especializadas en el campo de la imagen. Pero antes de entrar en detalle en el funcionamiento de esta, se ha creado el conjunto de imágenes o *dataset*, que permita entrenar la red a posterior. Todas las imágenes seleccionadas para el entrenamiento han sido tomadas desde el salpicadero de un vehículo, de forma que se simule la cámara que llevaría el vehículo autónomo propuesto.

Con el *dataset* ya creado y etiquetado, se ha entrado en detalle en la red neuronal convolucional, en especial en el funcionamiento del modelo y versión seleccionado, YOLO versión 5. Se ha elegido este modelo por su eficiencia y robustez en el análisis de imagen en video, dada la importancia que tiene este factor para el sistema. Además, al tratarse de un modelo ya funcional, empleando la transferencia de aprendizaje, se consigue que el sistema aprenda a detectar los objetos en menos tiempo.

Trabajando en conjunto con esta red se encuentra la segunda red neuronal, en este caso una red de regresión que analiza los tamaños de los objetos detectados por la red YOLO y decide si el vehículo debe frenar o no debido a la proximidad de estos. De nuevo en este caso ha sido necesaria la creación de un *dataset* para el entrenamiento de la red, de forma que pueda distinguir que objetos están lejos y cuales cerca.

Finalmente, el sistema completo con las dos redes operativas ha presentado unos buenos resultados tanto en la calidad y numero de aciertos en las detecciones como en la toma de decisiones sobre la acción que debe realizar el vehículo, frenando este cuando se acerca demasiado a un objeto. El número de *frames* por segundo que puede procesar es adecuado para un sistema que trabaje en tiempo real, dando un tempo de reacción suficiente para el correcto funcionamiento del sistema.

Así todo, existen numerosas posibilidades mediante las cuales se puede mejorar este sistema, algunas de ellas han sido ya esbozadas a lo largo del proyecto. A continuación, se entrará más en detalle en ellas de forma que estas presenten una línea de investigación y desarrollo futura para el proyecto:

- I. **Aumentar el conjunto de datos de entrenamiento.** Como se explicó al inicio del apartado tres, los conjuntos de datos para sistemas de conducción autónoma cuentan con miles o incluso decenas de miles de imágenes etiquetadas minuciosamente, buscando abarcar la mayor cantidad de casos posibles, mientras que este proyecto cuenta con solo unos cientos. A pesar de que solo con las imágenes del conjunto creado para el proyecto, ya se logran unas tasas de acierto muy altas y el vehículo funciona correctamente en la gran mayoría de situaciones, como se vio en el apartado de resultados, aun puede cometer errores que lleven al mal funcionamiento del vehículo. Esta es la razón del uso de conjuntos de entrenamiento tan grandes, que buscan una tasa de fallo prácticamente inexistente, haciendo el vehículo autónomo lo más seguro posible.
- II. **Control de los frenos.** Para implementar este sistema en un vehículo, es necesario que este pueda actuar sobre los frenos cuando se detecte un obstáculo, por seguridad el sistema de frenado actual del vehículo se mantendrá, por lo que el pedal del freno y el resto de componentes permanecerán en el vehículo. En cuanto a las opciones para el frenado autónomo, se puede emplear un actuador lineal conectado al cable de freno que traccionaría el mismo creando una tensión adecuada para ejercer una fuerza de frenado. También puede instalarse un actuador giratorio acoplado a una polea que recogería el cable creando la tensión necesaria para lograr un frenado progresivo.
- III. **Añadir más objetos para detectar.** Otra posible mejora, al igual que la anterior, parte del primer punto del desarrollo, en este caso etiquetando más objetos en las imágenes que forman el *dataset*, como podrían ser señales de tráfico, líneas viales, distinción entre tipos de vehículos (coches, camiones, furgonetas, etc.) o semáforos. Así el sistema podría adquirir una mejor comprensión del entorno al conocer más objetos. Esto llevaría a un nuevo nivel de inteligencia al sistema, pasando de solo acelerar y frenar a poder ser reactivo en función de los nuevos objetos detectados. Por ejemplo, si el vehículo aprende a reconocer las señales que delimitan la velocidad, este podrá adecuar la velocidad de su marcha según estas indiquen.
- IV. **Control de la dirección.** Este es un punto importante en el desarrollo de un vehículo autónomo ya que es el puente entre el sistema software de toma de decisiones y el propio vehículo. Existen varias posibilidades de realizar el control de la dirección, una de ellas es emplear actuadores electromecánicos en el brazo de la dirección, para poder actuar sobre las ruedas de una forma progresiva y poder transmitir el giro deseado a las mismas. Otra opción es emplear un motor-reductor conectado a un *encoder*, al cual se le transmitiría la orden de giro y este haría girar la columna de la dirección, pero este sistema presenta una limitación, solo puede hacer giros de 30°, debido a que si sigue girando el *encoder* realizaría una vuelta de mas y el sistema fallaría.

- V. **Añadir sensores complementarios.** El sistema propuesto cuenta solo con una cámara frontal, pero la simple inclusión de cámaras adicionales, que aprovecharían el software ya entrenado, permitiría al vehículo saber que tiene a los laterales o por detrás y con esta información estar más preparado para responder ante cualquier situación. Además de las cámaras, como ya se explicó en el segundo apartado, un vehículo autónomo puede disponer de una gran variedad de sensores como puedan ser un sistema LIDAR, sensores ultrasónicos o sensores infrarrojos, entre otros.
- VI. **Implementar el modelo en una placa.** Este modelo se ha probado únicamente con videos, el siguiente paso en el desarrollo sería la implementación de este software en una placa que, a su vez, pueda ser integrada en un vehículo de pruebas. De esta forma se podrá ver el funcionamiento en escenarios reales, donde el sistema además de las funciones ya estudiadas, podrá aplicar el resultado de la toma de decisiones de manejo sobre el propio vehículo.

Una placa donde podría instalarse este sistema en unas fases preliminares del desarrollo, podría ser la *Jetson Nano* de Nvidia, una placa especialmente diseñada para trabajo con sistemas de inteligencia artificial. Esta placa de solo 70x45 mm podría instalarse fácilmente en un pequeño vehículo de pruebas para comprobar el funcionamiento real del sistema completo. Esta placa cuenta con una potente CPU que ofrece 472 GFLOPs y una GPU de 128 núcleos, pudiendo ejecutar varias redes neuronales en paralelo y procesar varios sensores de alta resolución simultáneamente. Además, su consumo de potencia es muy bajo, entre 5 y 10 vatios.



Figura 45 Jetson Nano de Nvidia

6 Referencias

- [1] «Los niveles de la conducción autónoma». <https://www.ceaonline.es/blog/213-los-niveles-de-la-conduccion-autonoma>
- [2] «3 types of autonomous vehicle sensors in self-driving cars» <https://www.itransition.com/blog/autonomous-vehicle-sensors>
- [3] «Inertial measurement units will keep self-driving cars on track» <https://www.microcontrollertips.com/inertial-measurement-units-will-keep-selfdriving-cars-on-track-faq/>
- [4] «Redundancy for automated driving» /us/highlights/automatedmobility/redundancy-for-automated-driving/
- [5] L. Level 5, «Power Allocation Between Redundant Systems on Autonomous Vehicles», Medium, 04-abr-2019. <https://medium.com/lyftlevel5/https-mediumcom-lyftlevel5-power-allocation-between-redundant-systems-on-autonomousvehicles-a68f5d80f061>
- [6] «Self-driving vehicles: IoT makes roads safer», Ericsson.com, 05-sep2017. <https://www.ericsson.com/en/internet-of-things/trending/self-drivingvehicles>
- [7] J. Desmond, «Internet of Things (IoT) and AI Self-Driving Cars», AI Trends, 03-ago-2018. <https://www.aitrends.com/ai-insider/internet-of-things-iot-and-aiself-driving-cars/>
- [8] K. Burke, «How Does a Self-Driving Car See? | NVIDIA Blog», The Official NVIDIA Blog, abr. 16, 2019. <https://blogs.nvidia.com/blog/2019/04/15/howdoes-a-self-driving-car-see/>
- [9] R. Salas, «Redes Neuronales Artificiales». Departamento de Computación, Universidad de Valparaíso, 1998
- [10] L. Urresti, «Redes generativas para la reconstrucción de imágenes submarinas». Universidad de Cantabria, 2020
- [11] F. Izaurieta, C. Saavedra, «Redes Neuronales Artificiales». Departamento de Física, Universidad de Concepción, 2000
- [12] «Why Unsupervised Machine Learning is the Future of Cybersecurity», *TechNative*, ene. 28, 2020. <https://www.technative.io/why-unsupervised-machine-learning-is-the-future-of-cybersecurity/>
- [13] «Supervised vs Unsupervised vs Reinforcement Learning», Intellipaat Blog, dic. 26, 2019.
- [14] M. Massiris, C. Delrieux, A. Fernández, «Deteccion de equipos de protección personal mediante red convolucional YOLO». Actas de las XXXIX Jornadas de Automática, 2018
- [15] «¿Qué es una Red Neuronal Convolucional? Los OJOS de la Inteligencia Artificial - YouTube». <https://www.youtube.com/watch?v=V8j1oENVz00>
- [16] «Redes neuronales convolucionales CNN (Clasificación de imágenes) - YouTube». <https://www.youtube.com/watch?v=ns2L2T6wvAY>
- [17] P. Pastor Martín, «Usando Redes Neuronales Convolucionales Para Convertir Características Visuales en Estímulos Sonoros». Universidad de La Laguna, 2018

- [18] J. Torres, «Deep learning, introducción práctica con keras». Colección WATCH THIS SPACE, 2018
- [19] Tzutalin. Labellmg. Git code (2015). <https://github.com/tzutalin/labellmg>
- [20] «Training Data: What Is It? All About Machine Learning Training Data», *Appen*, abr. 14, 2020. <https://appen.com/blog/training-data/>
- [21] J. Redmon, S. Divvala, R. Girshick, y A. Farhadi, «You Only Look Once: Unified, Real-Time Object Detection», *arXiv:1506.02640 [cs]*, may 2016. Disponible en: <http://arxiv.org/abs/1506.02640>.
- [22] Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
- [23] P. Sierra Suarez, «Diseño e implementación de la dirección y frenos de un vehículo autónomo». Universidad Carlos III de Madrid

